File No. S360-28
Form C28-6517-0

4

IBM Systems Reference Library

IBM System/360 Operating System

Report Program Generator

This publication provides the information users of the IBM System/360 need to make use of the Report Program Generator.

The Report Program Generator provides programmers with an efficient, easy to use technique for generating programs that are capable of: (a) obtaining data records from input files, (b) performing calculations on data taken from those records or RPG literals, and (c) writing printed reports and other output files that consist of data taken from input records, the results of calculations performed by the program, and RPG literals.

This publication is divided into three sections. The first section covers such basic material as the capabilities and logic of the Report Program Generator. The second section covers all the information a programmer will need to use the Report Program Generator effectively. The third section covers six sample programs that make use of the Report Program Generator.

PROPERTY OF
THE BROWN UNIVERSITY COMPUTING LABORATORY

PREFACE

This publication provides the information users of the IBM System/360 need to make use of the Report Program Generator (hereinafter referred to as the RPG).

This publication is divided into three sections. The first section covers such basic material as the capabilities and logic of the RPG. The second section covers all the information a programmer needs to know to use the RPG. The third section covers six examples of programs written making use of the RPG.

A minimum of programming experience is required to use this publication.

NOTE: Terms pertinent to this publication are defined in the sections where they are used for the first time.

This publication was prepared for production using an IBM computer to update the text and to control the page and line format. Page impressions for photo-offset printing were obtained from an IBM 1403 Printer with a 120-character print chain containing upper and lower case letters, special characters, and numerals.

Copies of this and other IBM publications can be obtained through IBM Branch Offices.

A form has been provided at the back of this publication for readers' comments. If the form has been detached, comments may be directed to an IBM Systems Engineer or addressed to the IBM Corporation, Programming Systems Publications, Dept. D58, PO Box 390, Poughkeepsie, N.Y. 12602.

©1964 by International Business Machines Corporation

CONTENTS

INTRODUCTION..... 5
Purpose of the Program..... 5
Elements of the RPG..... 6
    The RPG Language..... 6
Capabilities of RPG Object Programs..... 7
Object Program Logic..... 8

USING THE RPG..... 10
RPG Language Entries..... 10
    Common Columns..... 11
    Comments Line..... 14
    File Definitions Entries..... 14
    Input Specifications Entries..... 16
    Calculation Specifications Entries..... 23
    Output-Format Specifications Entries..... 29
    Page Numbering..... 30
    Spacing Chart..... 30

EXAMPLES..... 37
    Input File..... 37
Program One (80/80 Listing)..... 38
Program Two (Detail Records)..... 44
Program Three (Final Total)..... 48
Program Four (Editing)..... 53
Program Five (Minor Totals)..... 56
Program Six (Header Records)..... 61

APPENDIX A..... 66
Table Lookup Operations..... 66

INTRODUCTION

Purpose of the Program

The RPG provides programmers with an efficient technique for writing source programs that can be translated into object programs capable of: (a) obtaining data records from input files (see Figure 1), (b) performing operations and calculations (e.g., comparison or addition) on data taken from those records or from RPG literals, and (c) writing a printed report (see Figure 2) and other output files. These output files may consist of data taken from input records, the results of operations and calculations performed by the object program, RPG literals, or any combination of the three. To make use of the RPG, the programmer does not have to be familiar with any other data processing program, or with the IBM System/360.

RPG Literal

An RPG literal is any group of numbers or characters, other than quote sign characters, that refers to and defines only itself and is entered on an RPG coding form. RPG literals that contain alphameric characters must be enclosed in quotation signs. RPG literals that contain nothing but numeric characters are not enclosed in quotation signs.

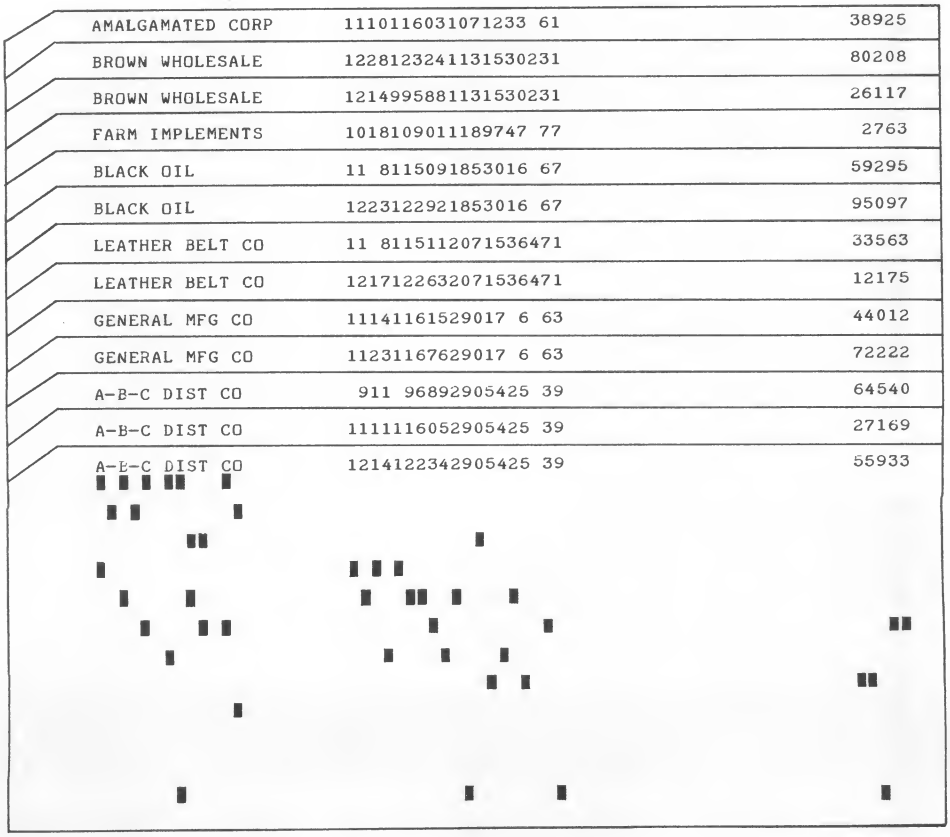

Figure 1. RPG - Input

(1) A C C O U N T S R E C E I V A B L E R E G I S T E R

(2) (3)	CUSTOMER NUMBER	CUSTOMER NAME	LOCATION STATE	LOCATION CITY	INVOICE NUMBER	INVOICE DATE MO	INVOICE DATE DAY	INVOICE AMOUNT
(4)	10712	AMALGAMATED CORP	33	61	11603	11	10	$ 389.25
(5)								$ 389.25*
(6)	11315	BROWN WHOLESALE	30	231	12324	12	28	$ 802.08
(7)	11315	BROWN WHOLESALE	30	231	99588	12	14	$ 261.17
(8)								$ 1,063.25*
(9)	11897	FARM IMPLEMENTS	47	77	10901	10	18	$ 27.63
(10)								$ 27.63*
(11)	18530	BLACK OIL	16	67	11509	11	8	$ 592.95
(12)	18530	BLACK OIL	16	67	12292	12	23	$ 950.97
(13)								$ 1,543.92*
(14)	20715	LEATHER BELT CO	36	471	11511	11	8	$ 335.63
(15)	20715	LEATHER BELT CO	36	471	12263	12	17	$ 121.75
(16)								$ 457.38*
(17)	29017	GENERAL MFG CO	6	63	11615	11	14	$ 440.12
(18)	29017	GENERAL MFG CO	6	63	11676	11	23	$ 722.22
(19)								$ 1,162.34*
(20)	29054	A-B-C DIST CO	25	39	9689	9	11	$ 645.40
(21)	29054	A-B-C DIST CO	25	39	11605	11	11	$ 271.69
(22)	29054	A-B-C DIST CO	25	39	12234	12	14	$ 559.33
(23)	.	.	.	.	.	.	.	$ 1,476.42*
(24)	.	.	.	.	.	.	.	$ 17,524.23**
	.	.	.	.	.	.	.	. .
	18 PRINT POSITIONS	32	59	67	79	88	97	109 111

Figure 2. RPG - Output

Elements of the RPG

The RPG consists of a source language and a compiler. The source language (i.e., the RPG language) provides programmers with a means for defining: (a) the overall or general characteristics of the files to which the input and output records belong, (b) the characteristics and fields of input data records, (c) RPG literals, (d) the operations or calculations that are to be performed by the object program, and (e) the characteristics and fields of the output records that are to be written on the printed report and other output files.

The compiler (i.e., the RPG compiler) is capable of taking the information specified by means of the RPG language and translating it into an executable object program.

The RPG Language

The RPG language is made up of the entries that may be specified on the RPG coding form. These entries are discussed in detail under "Using the RPG." The entries made on the form are transferred to punched cards and the cards

that are the result of this operation are brought together in one card deck. This card deck is the source program which the RPG compiler is to translate into an executable object program.

Capabilities of RPG Object Programs

The object program produced by the RPG compiler is capable of causing the execution of the operations discussed in the material that follows.

1. Write, at the top of each page of a printed report, lines (see "Line/Print Line" below) containing information that identifies the data printed on subsequent lines of the same page (see lines 1, 2, and 3 in Figure 2).

2. Obtain data records from input files.

3. Perform the following operations:

   a. Add the contents of a numeric field or the value of a numeric literal to the contents of another numeric field or the value of numeric literal and store the result in a result field (see "Record Field" and "Result Field" below).

   b. Set a specified result field to zeros, add the value of a literal or the contents of another field to the contents of the result field, and store the result in the result field.

   c. Subtract the contents of a numeric field or the value of a numeric literal from the contents of another numeric field or the value of another numeric literal, and store the result in a result field.

   d. Set a result field to zeros, subtract the value of a literal or the contents of another field from the contents of the result field, and store the result in the result field.

   e. Multiply the contents of a numeric field or the value of a numeric literal by the contents of another numeric field or the value of another numeric literal, and store the result in a result field.

   f. Divide the contents of a numeric field or the value of a numeric literal by the contents of another numeric field or the value of another numeric literal and store the quotient in a result field.

   g. Divide the contents of a numeric field or the value of a numeric literal by the contents of another numeric field or the value of another numeric literal and store the remainder in a result field.

   h. Compare the contents of a field or the value of a literal with the contents of another field or the value of another literal and set an indicator that specifies the result of the comparison.

   i. Move the contents of a field or the value of a literal into a result field.

   j. Search a table of arguments and functions, store the appropriate function (if any) in a result field, and set an indicator that specifies the result of the search.

4. Write data records on a printed report and/or other output files (see lines 4, 6, 7, etc., in Figure 2).

5. Write, on a printed report, lines that represent accumulations or totals taken from the data records printed on preceding lines of the report (see lines 5, 8, 10, etc., in Figure 2).

Record Field

A group of consecutive positions within a record: (a) that is to be treated as a logical whole, (b) that is a fixed number of positions in length, and (c) that has been assigned a symbolic label by the programmer.

Result Field

A special type of field. The primary function of this type of field is to provide an area in which the results of arithmetic operations can be stored.

Line/Print Line

One line of the standard print or listing form. For the purposes of this publication, each such line is considered one data record.

NOTE: The RPG is capable of reading input records that include fields that contain data in sterling formats and writing output records that include fields that contain data in sterling formats. The RPG is also capable of converting data contained in input record fields from sterling formats into pence, and the data contained in output record fields from pence to sterling formats. (All conversions include the signs of the relevant fields.) The sterling format of output record fields can be specified by use of an edit word. (See "Constant/Edit Word.") The RPG will also allow for the European conventions governing the use of decimal points and commas in printed reports.

Object Program Logic

The overall logic of the RPG is discussed here to provide the programmer with a general idea of the internal operation of RPG object programs.

RPG object programs perform the following operations:

1. If the object program is to produce a printed report, all print lines containing heading information specified by the programmer are written whenever the print form is at the beginning of a page.

2. One data record is obtained from each input file. If there is more than one input file the RPG obtains one data record from the input master file (see definition below) and one data record from each of the input support files (see definition below). The records from the master and each support file make up one logical unit. Each such logical unit is processed as if it were a single data record.

3. A check is made to determine whether a control break has occurred in any control field. (See "Control Break" and "Control Field" below.)

4. If a control break has occurred, any calculations that are to be executed as a result of the break are performed. In addition, if the object program is to produce a printed report, all print lines are written that represent totals the programmer has specified that the object program is to accumulate.

5. The calculations the object program is to perform under normal conditions (i.e., all calculations, the execution of which do not depend on a control break taking place), are executed. These operations are performed on data taken from RPG literals and/or data contained in the current input data record. The results of these operations are stored in the specified result fields.

6. One output data record is assembled for each output file the object program is to produce. The data included in these records is taken from: (a) the current input record, (b) result fields, (c) RPG literals or (d) any combination of the above.

7. The assembled data records are written on their respective output files.

8. Items 2-7 are repeated.

Control Field

A field, a change in the contents of which is used to determine whether the object program will perform special calculations that it would not otherwise perform, is termed a control field.

Control Break

A control break is said to have occurred in a control field whenever the contents of the field change. For example, consider the following situation:

1. Each record contained in a certain input field represents one purchase made by a customer of the reporting firm.

2. A particular field included in each record contains a number that identifies the customer who made the purchase.

3. The field discussed in item 2 is designated a control field.

Whenever a data record is obtained from the input file discussed in item 1, the customer identification number in that record is compared to the customer number contained in the record that preceded it. If the result of the comparison is unequal, a control break is said to have occurred in the customer number control field.

Master File

When data records are to be obtained from more than one input file, one of these files must be designated the master file. Generally speaking, the file designated the master file should be the input file that contains information of a permanent nature (e.g., in a payroll program; an employee's name, serial number, hourly wage, and department number).

Support Files

All files not designated the master file must be designated support files. Support files are distinguished from the master file by the nature of the information contained in their records. This information is generally of a variable nature. For example, in a payroll program, one support file might contain the number of regular hours and the number of overtime hours each employee has worked during the current week.

USING THE RPG

To make use of the RPG, the programmer must perform the steps outlined in the paragraphs that follow.

1. Describe, by means of the File Definitions entries made on the RPG coding form, the overall characteristics of the input files from which the object program is to obtain records, and the characteristics of the output files the object program is to produce.

2. Describe, by means of the Input Specifications entries made on the RPG coding form, the characteristics and fields of the records contained in each input file the object program is to process.

3. Define, by means of the Calculation Specifications entries made on the RPG coding form, the operations (e.g., add, subtract, move, etc.) the object program is to perform on data taken from input records and RPG literals.

4. Describe, by means of the Output-Format Specifications entries made on the RPG coding form, the characteristics and fields of the records the object program is to write on the printed report and other output files (if any).

5. Make up a source program card deck from the information entered on each RPG form; the information from one line punched in one card, the resulting cards arranged as shown in Figure 3.

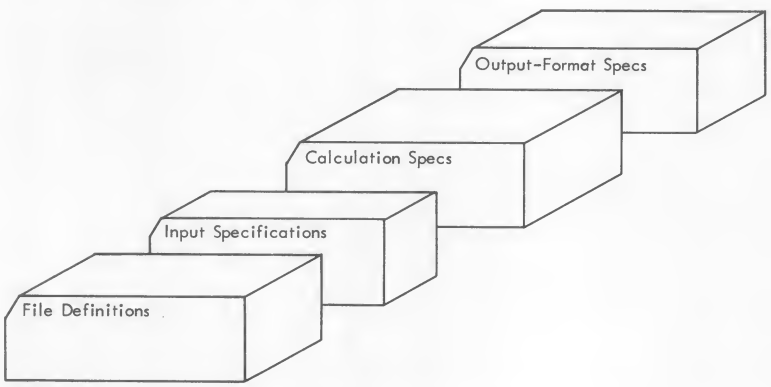

Figure 3. Source Program Card Deck

The object program produced by the RPG compiler from the source deck can then be executed to obtain the output files specified by the programmer.

RPG Language Entries

A detailed discussion of the entries that may be made on the RPG coding form (see Figure 4) is presented in this section of the publication.

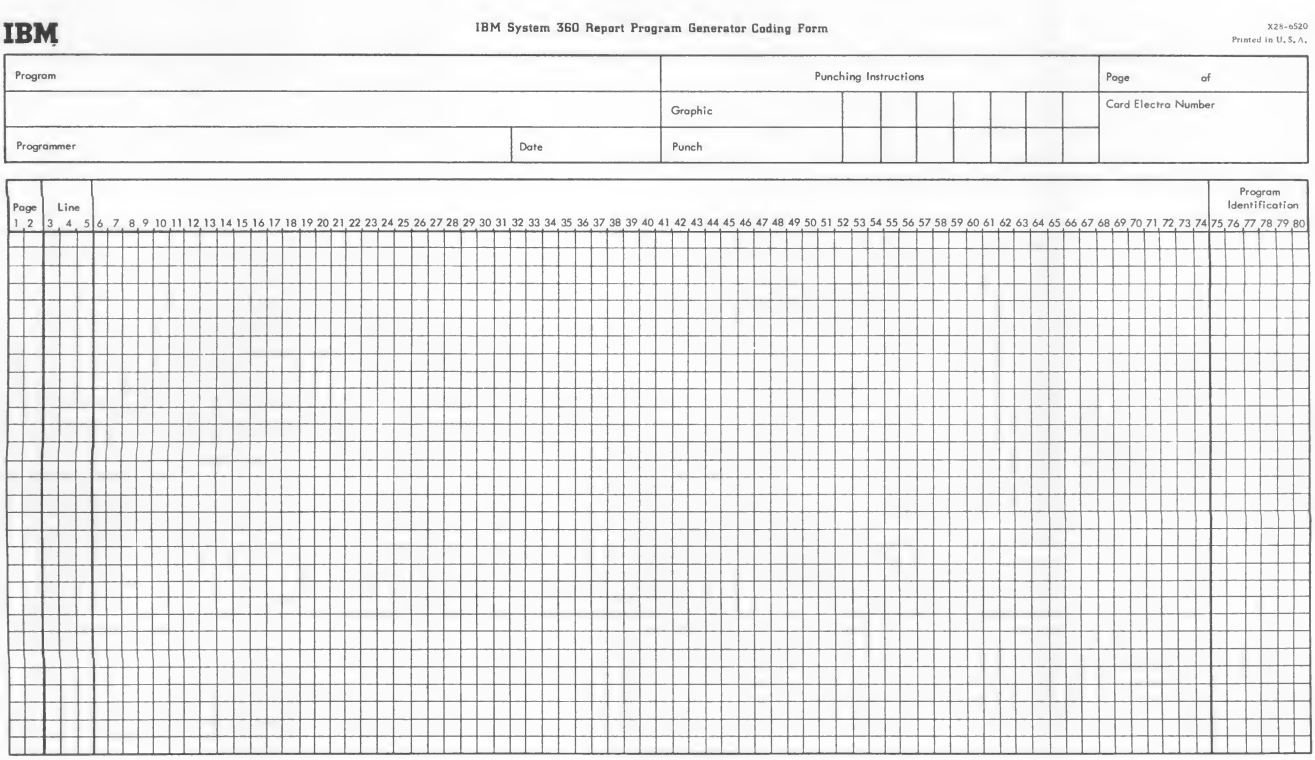

Figure 4. RPG Coding Form

RPG CODING CARD

The RPG coding card (see Figure 5) provides a convenient guide to assist programmers in making File Definitions, Input Specifications, Calculation Specifications, and Output-Format Specifications entries on the RPG coding form. The RPG coding card is essentially a pictorial breakdown of the columns in which the various entries are to be entered.

In order to use the card, the programmer determines what type of entries are to be made and places the portion of the card dealing with those entries above the line of the coding form on which the entries are to be made (see Figure 6).

Common Columns

The same type of information is entered in certain columns on all lines of the RPG coding form. These common columns are discussed in the material that follows.

Page Number (Columns 1-2)

Each page of the RPG coding form used by the programmer should be numbered consecutively starting with the number 01. The relevant page number should be entered in columns 1-2 of every line of the form on which the programmer makes an entry.

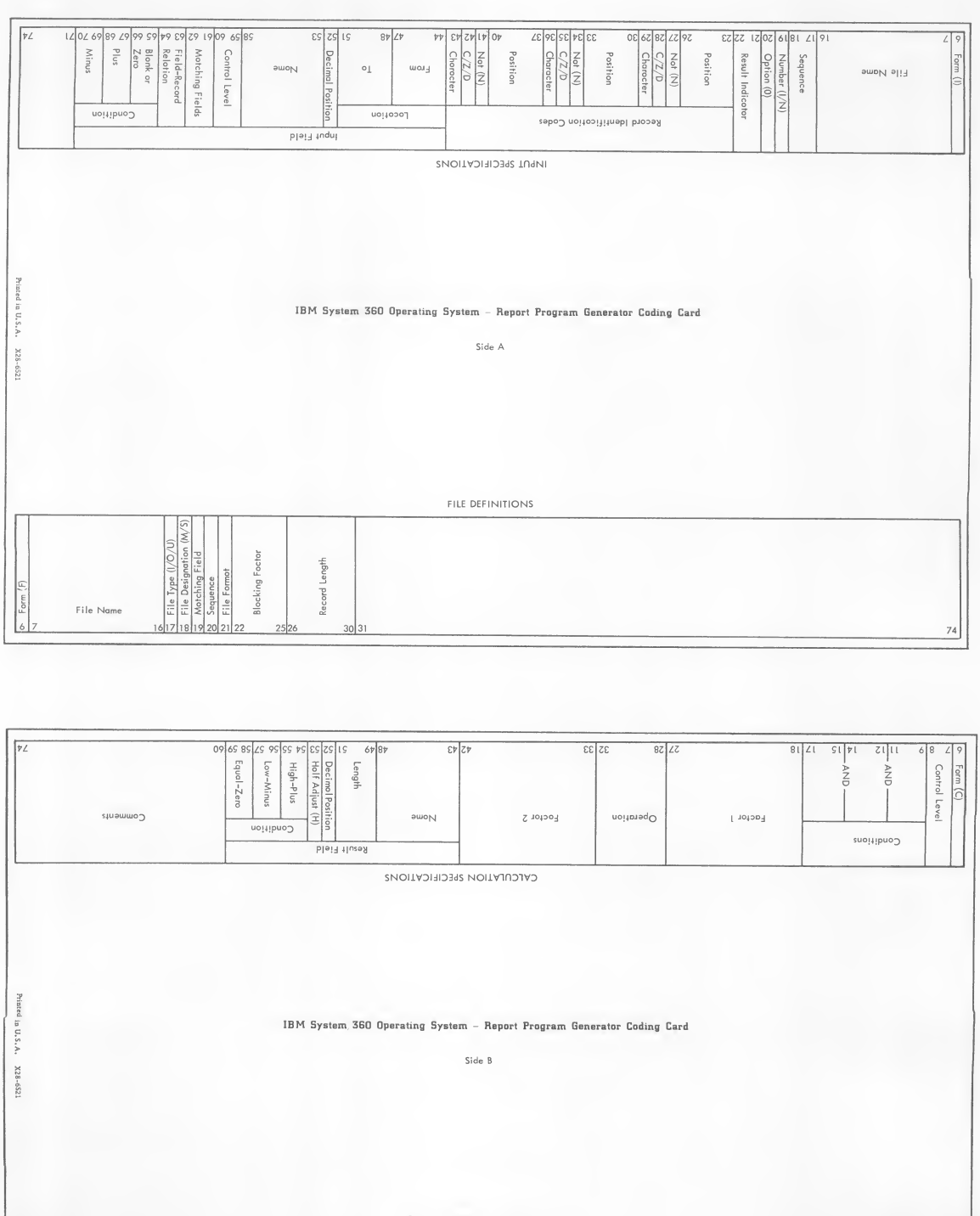

Figure 5. RPG Coding Card

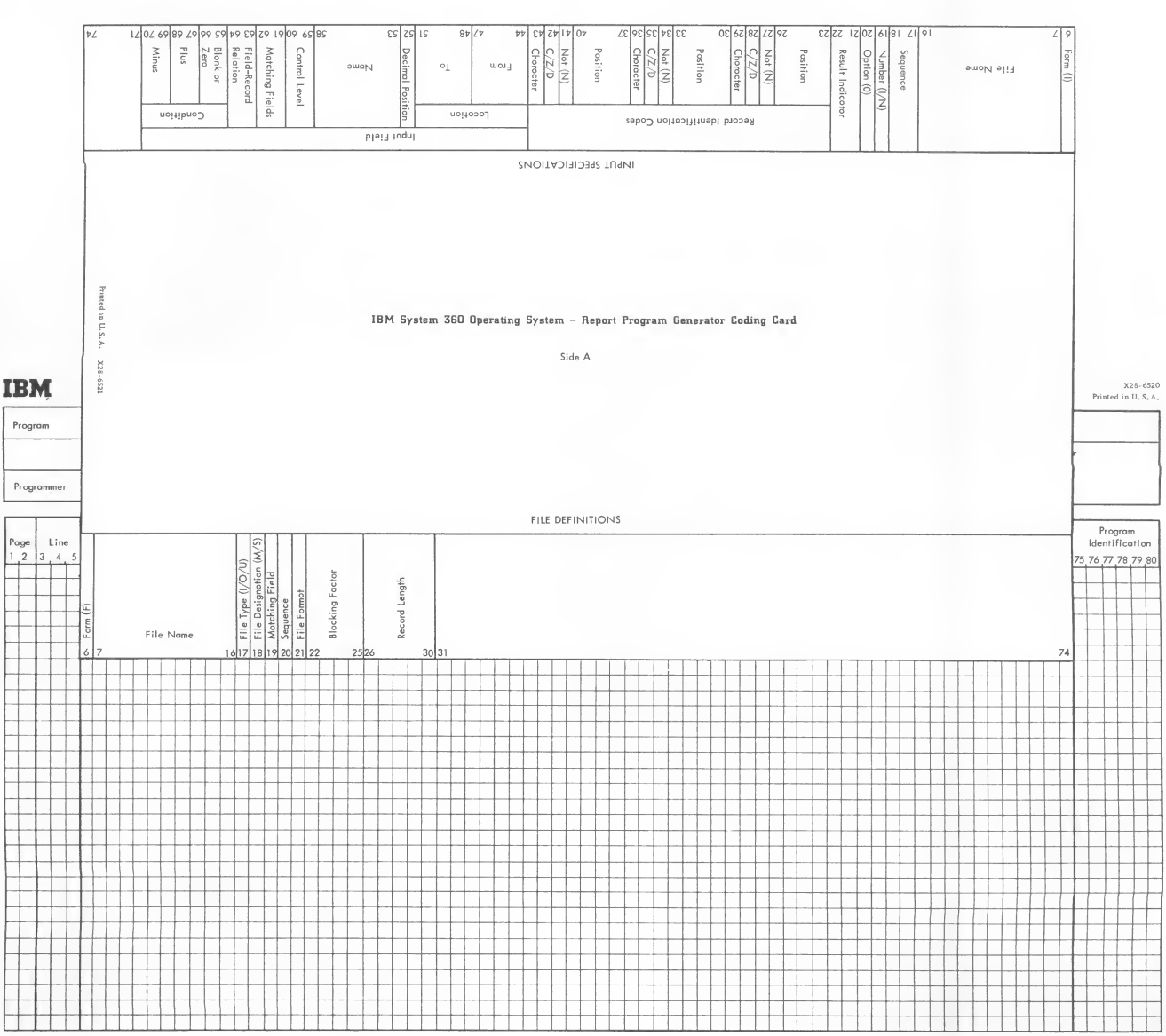

Figure 6. Using the RPG Coding Form and Card

Line Number (Columns 3-5)

A three-digit number should be entered in these columns of every line of the coding form on which the programmer has made an entry. The RPG compiler uses this number to perform a sequence check on the source program card deck. If a card is included in the deck out of sequence, an error message is written by the compiler.

Form Type (Column 6)

The character entered in this column indicates the type of entries that are being specified on the line, as follows:

CHARACTER	TYPE OF ENTRY
F	File Definitions
I	Input Specifications
C	Calculation Specifications
O	Output-Format Specifications

Program Identification (Columns 75-80)

The programmer may enter from one to six alphameric characters in these columns on each line to identify the program, or portion of a program, to which the entries made on the line belong.

Comments Line

If the programmer places an asterisk in column 7 of a line on the coding form, the RPG compiler assumes that the line contains only comments. The compiler will reproduce the information entered on such a line, unchanged, in the printed listing of the source program, in the same relative position in which the information was encountered in the source program. It will disregard the information on such a line in the production of the object program.

File Definitions Entries

The File Definitions entries define the characteristics of: (a) the input files from which the object program is to obtain data records, and (b) the output files on which the object program is to write data records. The entries made on one line define one file. Each file that is to be referenced by the object program must be defined.

File Name (Columns 7-16)

The name of the file defined on the line is entered, left-justified, in columns 7-16. The name of the file may consist of from one to ten alphabetic characters.

The file name entered in these columns must also be entered on the first line used to make the Input Specifications entries (for input files) or the Output-Format Specifications entries (for output files) that define the characteristics of the records that belong to the file.

File Type (Column 17)

The entry made in this column indicates the type of file defined on the line, as follows:

ENTRY	FILE TYPE
I	Input
O	Output
U	Update

Update Files

An update file is a file that is both an input and an output file. What makes it an update file is the way in which the object program handles the data records it obtains from and writes on the file. If the object program alters the data contained in one or more fields of each record belonging to the file without changing the nature of the data, the length of the field in which it appears, or the location of the field in each record, the file is considered an update file.

File Designation (Column 18)

If the object program is to reference two or more input files: (a) the letter M entered in this column indicates that the file defined on the line is the master file, and (b) the letter S entered in this column indicates that the file defined on the line is a support file. (Master files and support files are discussed under "Object Program Logic.")

No entry is made in this column if: (a) the file defined on the line is an output file, or (b) the object program is to reference only one input file.

Field Sequencing (Columns 19-20)

RPG object programs are capable of checking one specific field in each record obtained from the file defined on the line. This check can be used to determine whether the value of the data contained in the specified field is higher than, or lower than, the value of the data contained in the same field in the last record obtained from the file. Columns 19-20 will be used to indicate whether this check is to be made.

File Format (Column 21)

The entry made in this column indicates the type of records contained in the file defined on the line, as follows:

ENTRY	RECORD TYPE
1	Unblocked, variable-length records
2	Blocked, variable-length records
4	Blocked, fixed-length records

Blocking Factor (Columns 22-25)

When the number 1 has been entered in column 21, no entry is made in columns 22-25. If the number 2 has been entered in column 21, the entry made in columns 22-25 specifies the maximum number of records that may appear in each block. When the number 4 has been entered in column 21, the entry made in columns 22-25 specifies the number of records contained in each block.

Record Length (Columns 26-30)

If the number 1 is entered in column 21, no entry is made in columns 26-30. When the number 2 is entered in column 21, the entry made in columns 26-30 specifies the maximum number of characters that can appear in each block. If the number 4 is entered in column 21, the entry made in columns 26-30 specifies the number of characters in each record.

Columns 31-74

These columns are reserved for future IBM Programming Systems use.

Input Specifications Entries

The Input Specifications entries describe the characteristics and fields of the input data records that the object program must process to produce the desired printed report.

These entries are divided into two main groups, the input record definition entries and the input field definition entries. The input record definition entries describe the general or overall characteristics of the input data records. The input field definitions entries describe the makeup of the fields contained in the input data records.

When the records of a file are all of the same format, the programmer makes the input record definition entries on only one line of the form. When the records of a file are divided into two or more record groups, the programmer must make the input record definition entries on one line for each record group contained in the file (see "Input Record Groups" in the material that follows).

If the input record definition entries have been made by the programmer on a line of the coding form, no input field definition entries may be made on that same line. Conversely, if the input field definition entries have been made on a line, no input record definition entries may be made on the same line.

Only one record field may be defined on each line of the form. There is no restriction on the number of fields that may be defined.

The line used to describe the characteristics of the records must immediately precede the lines used to describe the fields found in each record.

Input Record Groups

The records of each input file may be divided into two or more groups.

If the input file is divided into record groups, each distinct group must be assigned an identifier. This identifier must be a unique two-character numeric or alphabetic code (e.g., 01, or AA).

The records of each group may appear on the relevant storage media in a particular sequence. That is, each record belonging to record group B appears on the storage media immediately following a record belonging to group A, and immediately preceding a record belonging to group C. Figure 7 shows card, tape, and disk files on which this type of organization has been imposed.

If a file is divided into record groups and the records belonging to the groups appear on the storage media in a predetermined sequence, the object program is capable of checking the order in which records are obtained from the file. In addition, it is capable of writing an error message and terminating the processing of the program if a record is obtained out of sequence.

The programmer indicates this check is to be performed by making no entry in column 20 of the line used to define the characteristics of the records belonging to each affected group. See "Option (Column 20)" for a detailed discussion of this feature.

When the programmer wants the object program to perform this check, the identifier assigned to each group must be a two-digit number that reflects the predetermined sequence. This means the identifier of one group must have a higher numeric value than the identifier of the group that precedes it in the sequence, and a lower numeric value than the identifier of the group that follows it.

If a file is divided into record groups, but the records of the groups do not appear in any particular order, or the order in which they appear is not to be checked by the object program, the identifier assigned to each record group must consist of two alphabetic characters.

The identifier assigned a particular record group is entered in columns 17-18 of the line of the form used to define the characteristics of the records belonging to the group. See "Sequence (Columns 17-18)" for a discussion of this entry.

INPUT RECORD DEFINITION ENTRIES

The input record definition entries are made in columns 7-43.

File Name (Columns 7-16)

The name of the file to which the records defined on the line belong must be entered, left-justified, in these columns. A file name may consist of from one to ten alphabetic characters. If a file is divided into two or more record groups, this entry is only made on the line used to define the first record group belonging to the file.

Sequence (Columns 17-18)

The identifier assigned to the records defined on the line, is entered in these columns. If the records belong to a record group, and they appear on

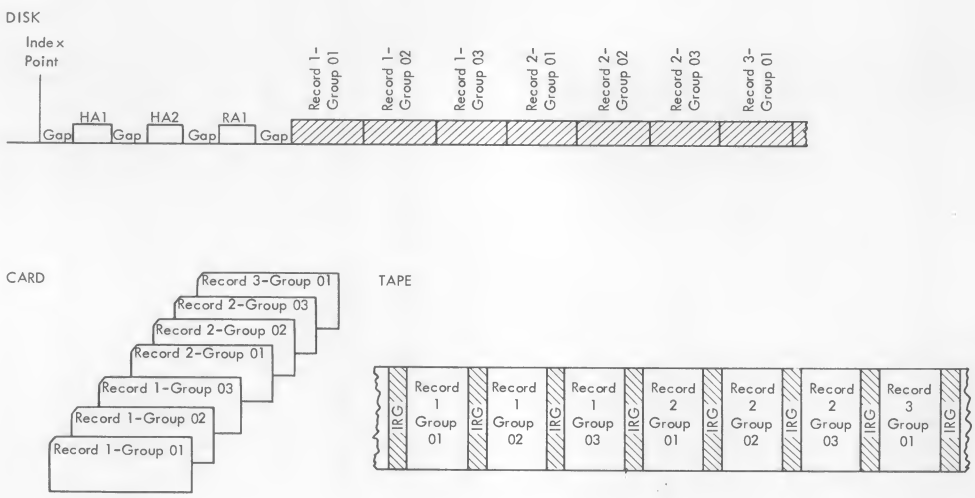

Figure 7. Predetermined Record Group Sequence

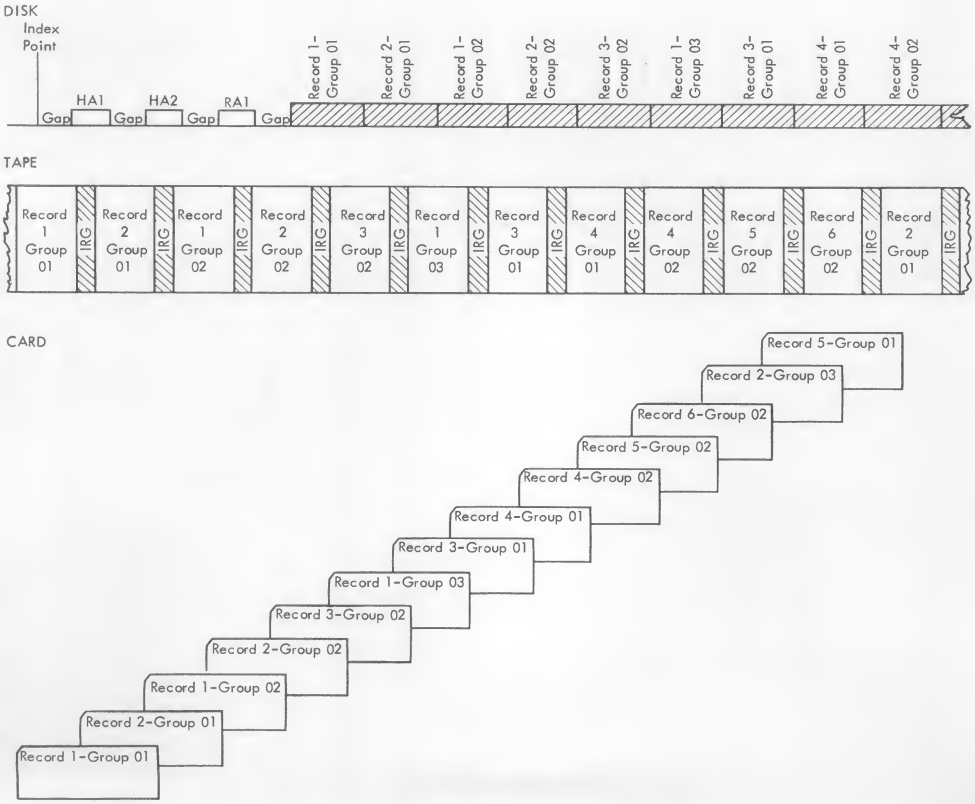

Figure 8. Record Group Set

the relevant storage medium in a predetermined order, the identifier must consist of two numeric digits. If the records do not belong to a record group, or belong to a group but the records of the group do not appear on the relevant storage medium in a predetermined sequence, the identifier must consist of two alphabetic characters.

All lines that contain two alphabetic characters in columns 17-18 must precede all lines that contain numeric characters in these columns.

Number of Records (Column 19)

If the records defined on the line belong to a record group and they appear on the relevant storage media one at a time (i.e., as shown in Figure 7), the number 1 is entered in this column. If the records belong to a record group and they appear on the relevant storage media two or more at a time (i.e., as shown in Figure 8), the letter N is entered in this column. If the records defined on the line do not belong to a record group (i.e., all records of the affected file have the same format) no entry is made in this column.

Option (Column 20)

No entry is made in this column, if the records defined on the line: (a) belong to a record group, (b) appear on the relevant storage medium in a predetermined order, and (c) the object program is to write an error message and discontinue processing whenever a record belonging to the group is obtained from the affected file out of sequence.

Under any conditions other than those outlined in the preceding paragraph, the letter O is entered in this column.

Result Indicator (Columns 21-22)

The programmer must enter, in these columns, the unique two-digit number that is to be assigned to the record available result indicator associated with the record group containing the records defined on the line. (Result indicators and record available result indicators are discussed below.) This entry may not be omitted.

Result Indicators

A result indicator is a switch within the object program. This is either ON or OFF. The fact that it is ON or OFF can be used to determine whether the object program will perform certain calculations the programmer has specified. (For a discussion of how a result indicator can be used in this way, see "Conditions" under the discussion of the Calculation Specifications entries.)

If such a switch is to be referenced by the object program, it must be labeled by the programmer. This label must be unique and must consist of two numeric characters.

Record Available Result Indicators

One record available result indicator is provided for each input record group the object program is to process. Whenever a record is made available to the

object program from that record group, the affected indicator is set ON. When the object program has completed processing the record, this indicator is set OFF.

Record Identification Codes (Columns 23-43)

From one to three record identification codes may be specified on each line.

If the programmer decides that more than three identification codes are required, additional codes may be entered on succeeding lines of the form. On each line used to specify additional codes, the code word AND must be entered, left-justified, in columns 17-19.

If the programmer wishes to specify identification codes that may be used instead of those entered on the initial and any succeeding lines, he may do this by: (a) entering the code OR, left-justified, in columns 17-19 of the next available line, and (b) specifying one or more identification codes on the same line. More than three such identification codes can be specified by entering AND in columns 17-19 of one or more lines following the line with OR entered in columns 17-19, and specifying additional codes on those lines.

Each record identification code is specified as follows:

1. The position, expressed as a four-digit number, in each data record of the character that is to be compared against the identification code, is entered in columns 23-26, 30-33, or 37-40 (whichever is appropriate).

2. The letter N is entered in column 27, 34, or 41 (whichever is appropriate) when the result of the comparison must be unequal for the affected record to be processed by the object program. If the result of the comparison must be equal, this column must be left blank.

3. The letter Z is entered in column 28, 35, or 42, if the zone portion of the identification code and the relevant character are to be compared; the character D, if the digit portions are to be compared; the character C, if both the zone and the digit portions (i.e., the complete character) are to be compared.

4. The character that is to be the record identification code is entered in columns 29, 36, or 43. This character may be any valid alphameric character.

Record Identification Code

A record identification code is a single alphameric character. This character, the zone portion of this character, or the digit portion of this character is compared against the character, or appropriate portion of the character, contained in a particular position within each record of the affected file or record group. This comparison is performed before the affected record is made available to the object program for processing. The result of this comparison, whether equal or unequal, may be used to determine whether the object program is to ignore the record or process it. If the programmer specifies that the result of this comparison must be equal, the affected record will not be processed if the result is unequal. Conversely, if the result must be unequal the object program will not process the affected record if the result is equal. There is no restriction on the number of record identification codes that the programmer can specify for one input file or record group.

INPUT FIELD DEFINITION ENTRIES

The input field definition entries are made in columns 44-70.

Input Field - From (Columns 44-47)

The position in each record, expressed as a four-digit number, of the first position of the field defined on the line, is entered in these columns.

Input Field - To (Columns 48-51)

The position in each record, expressed as a four-digit number, of the last position of the field defined on the line, is entered in these columns.

Input Field Decimal Position (Column 52)

If the field defined on the line contains numeric data, or alphameric data that is to be treated as if it were actually numeric data, a numeric digit (0-9) is entered in this column. The value of this digit represents the number of positions of data to the right of the implied decimal point contained in the data. If the data is a whole number, a zero is entered in this column. If the field contains alphameric data that is to be treated as alphameric data, this column is left blank.

Input Field Name (Columns 53-58)

The label that is to be assigned to the field defined on the line, expressed in alphabetic characters, is entered in these columns. Up to six characters are allowed and they must be left-justified.

Input Field Control Level (Columns 59-60)

If the field defined on the line is a control field, the control level of the field is entered in these columns. If the field is not a control field, these columns are left blank.

Control Levels

When data records contain more than one control field, it is possible to assign each such field a control level. This control level is expressed as a two-character code. The first character of this code must be the letter L. The second character must be a numeric digit (1-9). This digit indicates the control level of the field; the higher the numeric value of the digit, the higher the control level of the field. Whenever a control break occurs in a control field, all processing that is to be performed as a result of the break, is executed. (See "Control Level" in the discussion of the Calculation Specifications entries.) In addition, all processing is executed that is to be performed as a result of a control break in control fields that have been assigned control levels of a lower numeric value than the field in which the break actually occurred.

Matching Field Code (Columns 61-62)

No entry is made in these columns, unless the field defined on the line is a matching field. If the field is a matching field, the matching field code associated with the field must be entered in these columns.

Matching Fields

The concept of matching fields applies only when the object program will be required to process two or more input files (i.e., a master file and one or more support files). A matching field is a field contained in each record of one input file (e.g., the master file) that is the same length and contains the same information as a field contained in each record of each of the other input files (e.g., the support files). Each matching field must be assigned a matching field code. The first character of this code must be the letter M; the second must be the digit 1, 2, or 3. The object program compares each matching field in each record obtained from the master file with the analogous matching field in each record obtained from each of the support files. This means, for example, that the field designated M1 in each record of the master file is compared against the field designated M1 in each record of each support file.

Field-Record Relation (Columns 63-64)

When two or more input record types have similar formats, they can be specified by a single record type identifier using the OR relation (see discussion of record identification codes), with a unique record available result indicator assigned to each record type. For this purpose, the particular record available result indicator is entered in these columns to define the record type to which each field belongs. These columns are left blank when the fields are common to all such record types or when this facility is not being used.

Field Indicator (Columns 65-70)

The label that is to be assigned to the field indicator associated with the field defined on the line, is entered in these columns. The field condition the indicator is to represent when the indicator is ON, is determined by the columns the label is entered in, as follows:

FIELD CONDITIONS	LABEL ENTERED IN
Blank or Zero	65-66
Plus	67-68
Minus	69-70

Field Condition

The condition of a field containing alphameric data is said to be: blank, if the field contains only blanks; nonblank, if the field contains data other than blanks. The condition of a field containing numeric data is said to be

plus, if the contents of the field represent a positive number; minus, if the contents of the field represent a negative number; and zero, if the field contains only zeros, regardless of whether the sign of the field is plus or minus.

Field Indicators

A field indicator is a switch within the object program. It may be either ON or OFF, and the fact that it is ON or OFF can be used to determine whether the object program will perform certain operations on the data contained in the field with which the indicator is associated.

For a discussion of how a field indicator can be used in this way, see "Conditions", under the discussion of the Calculation Specifications entries.

If such a field indicator is to be checked by the object program, it must be labeled by the programmer. This label must be unique and must consist of two numeric digits. One field indicator may be associated with each field defined on the coding form.

Columns 71-74

These columns must be left blank. They are reserved for future IBM Programming Systems use.

Calculation Specifications Entries

These entries specify the operations the object program is to perform on data taken from input records and RPG literals. Each operation is specified on one line of the coding form and each operation must be specified in the order in which it is to be executed.

Control Level (Columns 7-8)

If a control break must occur in a control field for the object program to execute the operation specified on the line, the control level assigned to the affected control field must be entered in columns 7-8. If no control break is required, no entry is made in these columns.

The control level assigned to a control field is entered in columns 59-60 of a line used to make Input Specifications entries.

Conditions (Columns 9-17)

These columns may be used to specify the result indicators and/or field indicators that must be ON if the operation specified on the line is to be performed by the object program. A maximum of three result indicators, field indicators, or a combination of the two, may be specified.

The result indicators and/or field indicators that must be ON are specified by entering the labels of the relevant indicator(s) in columns 10-11, 13-14, and 16-17, Each of these labels must previously have been entered in: (a) columns 21-22 of a line used to make Input Specifications entries, in the case of record available result indicators; (b) columns 65-70 of a line

used to make Input Specifications entries, in the case of field indicators, or (c) columns 54-59 of a line used to make Calculation Specification entries, in the case of operation result indicators.

If a result indicator and/or field indicator specified in these columns must be OFF, the letter N is entered in column 9, 12, or 15 (whichever is appropriate).

Factor 1 (Columns 18-27)

The entry made in these columns is the label of a field, or an RPG literal that contains data to which the object program is to refer in performing the operation specified on the line. If the entry made in these columns is the label of a field, the field must be defined in: (a) columns 53-58 of a line used to make Input Specifications entries, or (b) in columns 43-48 of a line used to make Calculation Specifications entries. This label cannot exceed six characters in length and it must be left-justified.

If the entry made in these columns is a numeric Literal, it may consist of from one to ten characters. Included in these characters may be one decimal point to indicate the position of the decimal point in the literal, and one plus sign or one minus sign to indicate whether the literal contains a positive or a negative number. If a plus sign or a minus sign is included, it must appear as the leftmost character of the literal. Numeric literals must be left-justified.

If the entry made in these columns is an alphameric literal, it may consist of from one to eight alphameric characters (except quotation signs) and these characters must be enclosed by quotation signs. Alphameric literals must be left-justified in the block and they may not be used in arithmetic operations.

If no entry is made in these columns, an error message will be printed, unless the relevant operation does not refer to factor 1 (e.g., Z-ADD).

Operation Code (Columns 28-32)

The operation that the object program is to perform is indicated by the operation code entered in these columns. There are eleven operation codes. They are as follows:

ADD (Add)

This operation code causes the contents of the field or literal specified in factor 1 to be added, algebraically, to the contents of the field or literal specified in factor 2. The result of this operation is placed in the result field specified on the line. The field(s) and/or literal(s) involved in this operation can only contain numeric characters.

Z-ADD (Zero and Add)

This operation code causes the data contained in the numeric literal or field specified in factor 2, to be placed in the result field specified on the line, after that result field has been set to zeros. Factor 1 is not referenced during this operation.

SUB (Subtract)

This operation code causes the contents of the field or literal specified in factor 2 to be subtracted, algebraically, from the contents of the field or literal specified in factor 1. The result of this operation is placed in the result field specified on the line. The field(s) and/or literal(s) involved in this operation can only contain numeric characters.

Z-SUB (Zero and Subtract)

This operation code causes the negative of the number contained in the literal or field specified in factor 2 to be placed in the result field specified on the line. This operation is performed after the indicated result field has been set to zeros. The field or literal specified in factor 2 can only contain numeric characters. Factor 1 is not referenced during this operation.

MULT (Multiply)

This operation code causes the contents of the field or literal specified in factor 1 to be multiplied, algebraically, by the contents of the field or literal specified in factor 2. The result of this operation is placed in the result field specified on the line. The field(s) and/or literal(s) involved in this operation can only contain numeric characters.

DIV (Divide)

This operation code causes the contents of the field or literal specified in factor 1 to be divided by the contents of the field or literal specified in factor 2. The result of this operation (i.e., the quotient) is placed in the result field specified on the line. The contents of the field or literal specified in factor 2 cannot be equal to zero. The field(s) and/or literal(s) involved in this operation can only contain numeric characters. Any remainder that results from this operation will be lost.

DIVR (Divide With Remainder)

This operation code causes the object program to perform the same operations that the DIV operation code causes it to perform, with one exception. The exception is that the remainder resulting from the operation, rather than the quotient, will be placed in the result field specified on the line. If no remainder results from the operation the relevant result field will be set to zeros.

COMP (Compare)

This operation code causes the contents of the field or literal specified in factor 1 to be compared against the contents of the field or literal specified in factor 2. The outcome of this operation may be used to turn an operation result indicator ON (see "Result Field Conditions").

If the field or literal specified in factor 1 contains numeric data and the field or literal specified in factor 2 contains alphameric data, only the numeric or digit portions of the data contained in the latter is involved in the comparison. The converse of this statement is also true. When the field(s) and/or literal(s) involved in this operation both contain alphameric

data, the operation is performed according to the internal collating sequence.

If the field(s) and/or literal(s) involved both contain alphameric data and are not equal in length, the shorter of the two is padded with low-order blanks until it is the same length as the larger. If the field(s) and/or literal(s) involved contain numeric data and are not equal in length, they are aligned according to the position of the implied decimal point contained in each. The shorter field or literal is then padded with high-order and low-order zeros until it is the same length as the larger.

MOVE (Move)

This operation code causes the contents of the field or literal specified in factor 2 to be placed in the result field specified on the line. The data moved by this operation is placed in the affected result field. The field or literal specified in factor 2 may contain alphameric or numeric data. Factor 1 is not referenced by this operation.

MZONE (Move Zone)

This operation code causes the zone portion of the low-order character contained in the field or literal specified in factor 2 to be moved to the low-order character of the indicated result field. Factor 1 is not referenced by this operation.

LOKUP (Table Lookup)

This operation code causes a table lookup operation to be performed. The field specified in factor 1 contains the first table argument of the affected table. The field or literal specified in factor 2 contains the relevant search argument. The table function located by the operation (if any) is placed in the result field specified on the line. The results of this operation may be used to set an operation result indicator ON (see "Result Field Conditions").

Factor 2 (Columns 33-42)

The entries, the purpose of the entries, and the rules governing the entries, that may be made in factor 2 are the same as those detailed in the discussion of factor 1. If no entry is made in these columns, an error message will be printed unless the relevant operation does not refer to factor 2.

Result Field Name (Columns 43-48)

The label of the result field that is to be associated with the operation specified on the line, is entered, left-justified, in these columns. Such a label may consist of from one to six alphabetic characters.

If the operation specified on the line is a table lookup operation and (a) the table does not have functions associated with the table arguments, or (b) functions are associated with the arguments but the programmer does not want the relevant function placed in a result field, columns 43-48 may be left blank.

Result Field Length (Columns 49-51)

The length of the result field defined on the line, expressed as a three-digit number, is entered in these columns. If the length of the data placed in this field exceeds the length of the field, the high-order characters of data are lost.

Result Field Decimal Position (Column 52)

When numeric data is to be placed in the result field defined on the line, a one-digit number is entered in this column that is equal to the number of positions of data the programmer wishes to retain that are to the right of the implied decimal point contained in the data. If alphameric information is to be placed in the result field, no entry is made in this column.

Result Field Half Adjust (Column 53)

The letter H entered in this column causes a numeric value of 5 to be added to the first character to the right of the last character the programmer wishes to retain in the data that is to be placed in the result field. The character to which 5 was added and all characters to the right of it are then dropped and the balance of the data is placed, right-justified, in the result field. If the object program is not to perform this operation, or the data that is to be placed in the result field is alphameric data, no entry should be made in this column.

Result Field Conditions (Columns 54-59)

Three different result indicators (i.e., operation result indicators) may be labeled in these columns. The indicators and the conditions under which each will be set ON are shown in Figure 9.

If the operation defined on the line is a table lookup operation, the result indicator(s) that the programmer labels in columns 54-59, perform an additional function. In this situation, they indicate the type of search that is to be executed. The manner in which they do this is discussed in the paragraphs that follow.

When the table arguments appear in storage in random order, (see "Random Order" below) the only type of search that may be performed is a search for a table argument that is equal to the search argument. Under these conditions, the programmer must enter a label in columns 58-59, but he may not enter a label in columns 54-55 or 56-57. The fact that a label appears in columns 58-59, but none appear in columns 54-57, indicates a search for a table argument equal to the search argument, is to be performed.

When the table arguments appear in storage in an ascending or descending sequence (see "Ascending Sequence" and "Descending Sequence" below), the types of searches that may be performed and the indicators that must be labeled to identify the relevant type of search, are shown in figure 10.

Random Order

A table of arguments is said to be organized in random order when the arguments do not appear in storage according to any order or sequence.

INDICATOR	INDICATOR SET ON		
Label Entered In Columns	Arithmetic Calculation	Compare	Table lookup
54-55	Data placed in result field is a plus number	Value of the contents of the field or literal in factor 1 is higher than that of the field or literal specified in factor 2	Smallest table argument higher than search argument is found
56-57	Data placed in result field is a minus number	Value of the contents of the field or literal in factor 1 is lower than that of the field or literal specified in factor 2	Greatest table argument lower than search argument is found
58-59	Data placed in result field is equal to zero	Value of the contents of the field or literal in factor 1 is equal to that of the field or literal specified in factor 2	Search argument equal to an argument found in the table

Figure 9. Operation Result Indicators

TYPE OF SEARCH	COLUMNS INTO WHICH RESULT INDICATOR LABEL MUST BE ENTERED
Search for smallest table argument higher than search argument	Columns 54-55
Search for greatest table argument lower than search argument	Columns 56-57
Search for table argument equal to search argument	Columns 58-59
Search for table argument higher than, or equal to, search argument	Columns 54-55 and Columns 58-59
Search for table argument lower than, or equal to, search argument	Columns 56-57 and Columns 58-59

Figure 10. Table Lookup

Ascending Sequence

A table of arguments is said to be organized in an ascending sequence when the value of a table argument is higher than the value of the argument that precedes it, and lower than the value of the argument that follows it.

Descending Sequence

A table of arguments is said to be organized in a descending sequence when the value of a table argument is lower than the value of the argument that precedes it, and higher than the value of the argument that follows it.

Comments (Columns 60-74)

Columns 60-74 constitute the comments columns. The programmer may enter any comments he wishes in these columns.

Output-Format Specifications Entries

The Output-Format Specifications entries define the characteristics and fields of the data records that are to be written on the printed report and other output files (if any) the object program is to produce.

The Output-Format Specifications entries are divided into two main groups. These groups are the output record definition entries, and the output field definition entries.

The output record definition entries made on one line of the form are used to define the characteristics of the records belonging to a particular record group of an output file or of a particular output file, if that file is not divided into groups of records.

When the programmer makes the record definition entries on a line of the form, no field definition entries may be made on that same line.

The field definition entries are used to define each field contained in the records belonging to a particular output record group or file. Only one field may be defined on each line. The lines of the form used to define the fields of the records belonging to a particular file or group should immediately follow the line used to define the characteristics of the records belonging to that file or group. If the programmer makes the field definition entries on a line of the form, no record definition entries may be made on that same line.

Output Record Groups

Each output file may be divided into two or more groups of records. The records belonging to each group may then have a different format. This allows the programmer to mix records of different formats on the same output file. If an output file is divided into record groups, the record definition entries made on one line of the coding form must be used to define the characteristics of the records belonging to one group.

Page Numbering

When an output file is to be written on a printer, the object program is capable of causing consecutive page numbers to be printed on each page of the print form used for the file. If the object program is to perform this function, the record definition entries must be made on a line of the form as if each page number were a field of a data record belonging to a particular record group of the printer output file. In addition, certain special information must be supplied by the field definition entries made on the next line of the form. The special information that must be provided in these columns is detailed in the appropriate sections of the discussion covering the field definition entries.

Spacing Chart

A spacing chart is provided to allow the programmer to lay out a pictorial representation of the format of each output file that the object program is to produce. This chart was designed for printer output files, but it may be used effectively for other types of output files. The numbers across the top and the bottom of this chart represent the print positions available on the standard print or listing form. The numbers down the left-hand edge of the chart indicate the lines that are available on each page of the standard print form. This chart is especially useful for establishing the print positions in which the records of a file are to be printed and the spacing that is to exist between each record. An example of the spacing chart is shown in Figure 11.

In the example shown in Figure 11, all constant information is placed in the print positions in which it is to appear; variable information is represented by x's; special characters are entered where they are to appear; and zero suppression is indicated by a zero placed in the position where the zero suppression operation is to end.

OUTPUT RECORD DEFINITION ENTRIES

The record definition entries are made in columns 7-33.

File Name (Columns 7-16)

The name of the file to which the records defined on the line belong must be entered, left-justified, in these columns. A file name may consist of from one to ten alphabetic characters. If a file is divided into two or more groups, this entry must be made on each line used to define a record group belonging to the file.

Output Record Type (Column 17)

Output files may contain three different types of records. The three types are header records, detail records, and total records (see definitions below).

The letter H entered in column 17 indicates that the records defined on the line are header records. The letter D entered in column 17 indicates that they are detail records. The letter T entered in column 17 indicates that they are total records. There must always be an entry made in this column.

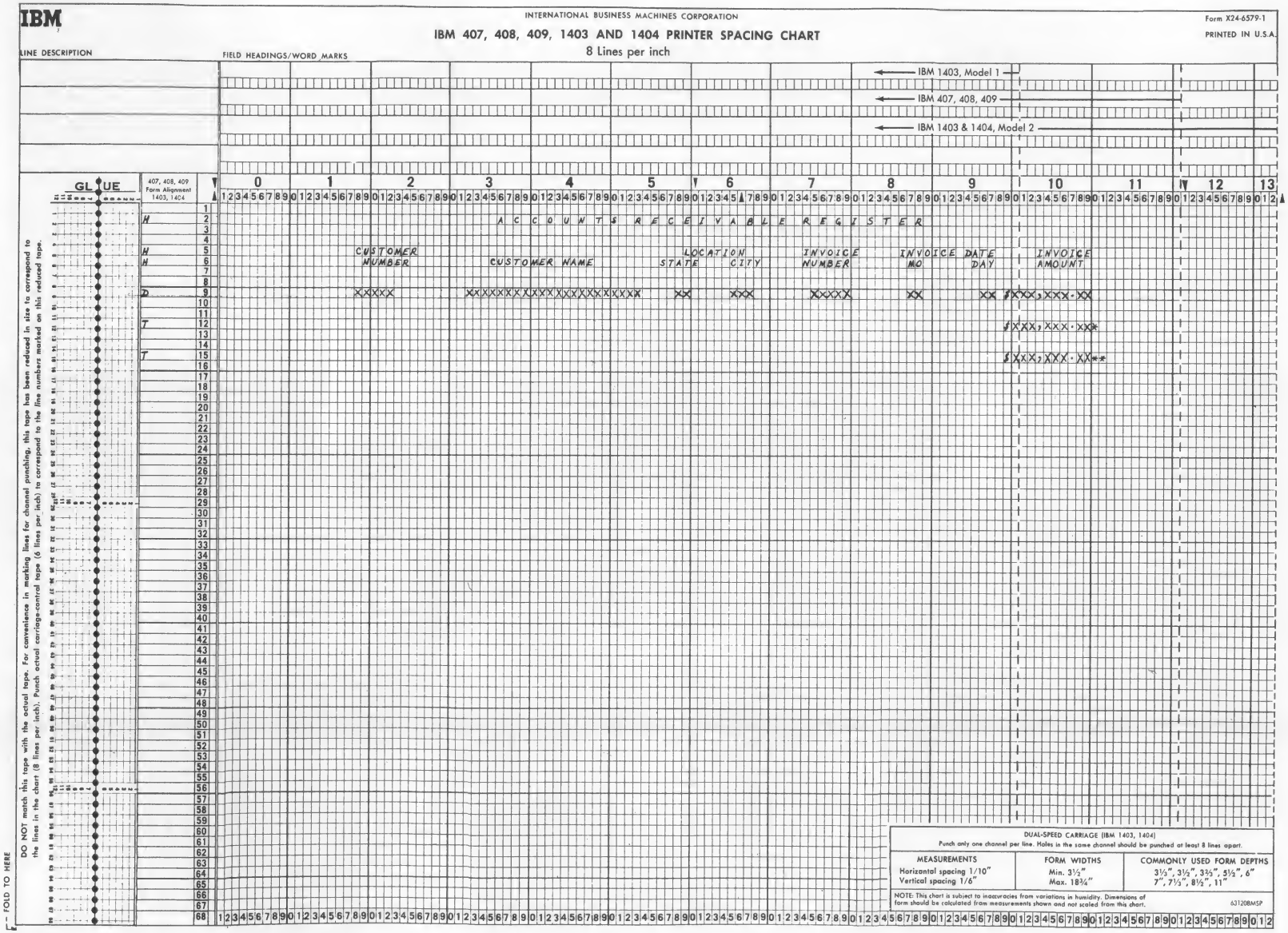

Figure 11. Printer Spacing Chart

Header Record

A header record is a data record containing information that serves to identify other data included in the same file. The records shown on lines 2, 5, and 6 of the spacing chart illustrated in Figure 11 are header records. Each of the header records shown in Figure 11 belongs to a different record group.

The information that is to be contained in each header record belonging to a particular record group may be made available to the object program in several ways. One method is covered under "Constant/Edit Word" in the discussion of the output field definition entries.

Once the information that is to be contained in each header record belonging to a particular record group has been made available, it is stored so that the object program may reference it whenever a header record belonging to that group is to be written on the output file.

Each time a record is written on the printer output file, the object program makes a check to determine if a header record belonging to any of the record groups included in the file is to be written on the file at that time. If a header record from each of two or more groups are to be written at that time, they are written in the order in which their characteristics are defined on the form.

Detail Record

A detail record is a record made up of data taken from input records, the results of operations performed by the object program, or RPG literals. An example of a detail record is a record that contains the type of information shown on line 9 of the spacing chart illustrated in Figure 11.

Total Record

A total record is a record made up of data that is a total or accumulation of some kind. Examples of total records are the records shown on lines 12 and 15 of the spacing chart illustrated in Figure 11. The record shown on line 12 represents a minor total (i.e., an accumulation of all the invoice amounts due from a particular customer). The record shown on line 15 represents a final total (i.e., the accumulation of all the invoice amounts due from all the customers of the reporting firm). Each of the total records shown in Figure 11 belongs to a different record group.

Whenever a record is written on the printer output file, the object program makes a check to determine if a total record belonging to any record group included in the file is to be written on the file at that time. If a total record from each of two or more record groups are to be written at that time, they are written in the order in which their characteristics are defined on the coding form. (Control breaks may also be used to force the printing of total records.)

Stacker (Column 18)

No entry is made in this column unless the records of the file or record group defined on the line are to be punched on a card-punch output device.

The one-digit number entered in this column indicates the stacker into which punched cards belonging to the relevant file or record group are to be selected.

Printer Spacing (Columns 19-20)

No entries are made in these columns unless the records of the file or record group defined on the line are to be written on a printer. The one-digit number (1-9) entered in column 19 indicates the number of lines that the print form is to be spaced before each record is printed. The one-digit number placed in column 20 indicates the number of lines that the print form is to be spaced after each record has been printed. If the print form is to be spaced normally before and/or after printing each record, no entries are made in these columns.

Printer Channel Skipping (Columns 21-24)

No entries are made in these columns unless the records defined on the line are to be written on a printer. The two-digit number entered in columns 21-22 indicates the control carriage tape channel that the printer is to skip to before each of the affected records is printed. The two-digit number entered in columns 23-24 indicates the control carriage tape channel that the printer is to skip to after each record is printed. If the printer is not to skip control carriage tape channels before and/or after printing each record, no entries are made in these columns.

Output Conditions (Columns 25-33)

The entries made in these columns (if any) specify the conditions under which each of the records defined on the line is to be written. The programmer may enter the labels of from one to three result indicators or field indicators in these columns; one in columns 26-27; one in columns 29-30; and one in columns 32-33. If the specified indicator must be ON for a record of the relevant file or record group to be written, the column (i.e., column 25, 28, or 31) immediately preceding those the indicator was specified in, must be left blank. If the specified indicator must be OFF, the character N must be placed in column 25, 29, or 31 (whichever is appropriate).

NOTE: The programmer may specify additional or alternate result indicators or field indicators in the same way that additional or alternate record identification codes are specified. See "Record Identification Codes" under the description of the Input Specifications entries.

The indicators specified in these columns may include: (a) any result indicator specified in columns 21-22 of a line used to make Input Specifications entries, (b) any result indicators specified in columns 54-59 of a line used to make Calculations Specifications entries, (c) any of three special indicators (see "Special Indicators" below), and (d) any field indicator specified in columns 65-70 of a line used to make Input Specifications entries.

Special Indicators

There are three special indicators. They are the first page special indicator, the page overflow special indicator, and the last record special indicator. Each of these indicators has been assigned a label that must be used by the programmer if he wishes to refer to the indicator.

The first page special indicator cannot be used unless the records defined on the line are to be written on a printer. This indicator is set OFF when the printer spaces the print form to the second page of the output file. It is ON until that time and the label assigned to this indicator is 1P.

The page overflow special indicator cannot be used unless the affected records are to be written on a printer output device. This indicator is set ON whenever the end of a page of the print form is reached. This indicator is set OFF as soon as the print form is spaced to the next page. The label assigned to this indicator is OF.

The Last Record Special Indicator is set ON: (a) when the last record is obtained from the input file, if there is only one input file, or (b) when the last record is obtained from the master file, if there is more than one input file. This indicator is OFF until that time and the label assigned to this indicator is LR.

OUTPUT FIELD DEFINITION ENTRIES

The output field definition entries are made in columns 34-70.

Output Field Name (Columns 34-39)

The label of the field defined on the line is entered, left-justified, in these columns, unless this field is to contain an RPG literal (see "Constant/Edit Word"). The label entered may consist of from one to six alphameric characters. This label must be a label that has previously been assigned to an input record field or a result field. That is, the label must have previously been entered in columns 53-58 of a line used to make Input Specifications entries, or in columns 43-48 of a line used to make Calculation Specifications entries.

NOTE: The code word PAGE must be entered, left-justified, in these columns, if the line is to provide the special information necessary for the object program to print a page number record on each page of the print form used for a printer output file. If the page number is to be reset to 1 under certain conditions, those conditions are specified by the entries made in columns 25-33 of the line with PAGE entered in columns 34-39.

Zero Suppression (Column 40)

The letter Z entered in this column causes a zero suppression operation to be performed on the contents of the field defined on the line. This operation will be performed before each record containing the field is written. If an edit word or RPG literal is entered in columns 46-70 of the line, or zero suppression is not wanted, no entry is made in this column.

Blank After (Column 41)

The letter B entered in this column indicates that the contents of the field defined on the line are to be altered each time a record containing the field is written on the appropriate output file. If the field contains alphameric data, it is set to blanks. If the field contains numeric data, it is set to zeros.

Output Field End Position (Columns 42-45)

The number entered, right-justified, in these columns indicates the position of the last character contained in the field defined on the line within each record that includes the field. When the field is the last in each record that contains it, the number entered in these columns must equal the number of character positions contained in each record.

NOTE: If the line is being used to provide special information necessary for the object program to number each page of a printer output file, the number entered in these columns indicates the position on the line of each print form page in which the low-order digit of the page number is to be printed.

Constant/Edit Word (Columns 46-70)

An RPG literal of up to twenty-three alphameric characters may be entered in these columns. If a literal is entered it must be enclosed by quotation signs ('), even if it contains nothing but numeric characters. The RPG compiler will take this literal and place it, right-justified, in the field defined on the line. This information will then be output in each record that contains the field.

NOTE: Entering a literal in these columns is a convenient way of providing the constant information that is to be included in header records.

The programmer may enter an edit word, enclosed in quotation signs ('), in these columns, rather than a literal. If an edit word is entered, the object program will reference it to edit the contents of the field defined on the line. This editing operation will be performed before each record containing the field is written on the relevant output file.

When a literal is entered in columns 46-70, columns 34-39 (i.e., field name) on the same line must be left blank. When an edit word is entered, column 40 (i.e., zero suppression) on the same line must be left blank. (The editing operation is capable of performing zero suppression. The zero entered in the units position of the edit word shown in Figure 12 caused the supression of all zeros encountered in the data field up to and including the *implied* units position. If an asterisk had been placed in this position, each zero that was suppressed would have been replaced with an asterisk.)

Edit Word

An edit word is a field made up primarily of special characters. These special characters are used to control the editing of a data field, or are themselves inserted in the data field in the same relative position that they appear in the control word. An example of the editing process is shown in Figure 12.

EDIT WORD	$	b	b	b	,	b	b	0	.	b	b
UNEDITED DATA	0	0	2	5	7	4	2	6			
EDITED DATA	$			2	,	5	7	4	.	2	6

Figure 12. Edit Operation

Columns 71-74

These columns are left blank. They are reserved for future IBM Programming Systems use.

EXAMPLES

The generation of six object programs, each of which is capable of producing a printed report (i.e., printer output file) of varying degrees of detail and complexity, is discussed in this section of the publication.

The first object program discussed is an 80/80 listing. That is, an object program capable of reading data records from a punched card input file, and writing those same data records on a printer output file, without altering or operating on the data contained in the records.

The second object program described in this section is capable of reading the same input records discussed above; rearranging the order of, and altering the spacing between, the fields contained in the records; and writing, as detail records, the reorganized data records on a printer output file.

The third object program covered, in addition to the functions performed by the second, is capable of accumulating data for a total record that represents a final total.

The fourth object program presented here, in addition to the functions performed by the third object program, is capable of causing an edit operation to be performed on the contents of a field in each detail record and on the contents of the total record.

The fifth object program, in addition to the functions performed by the fourth object program, is capable of accumulating and printing total records that represent minor totals.

The sixth object program, in addition to the functions performed by the fifth object program, is capable of writing certain specified header records at the beginning of each page of the print form used for the printer output file.

Input File

Each object program discussed in this section references one input file. This input file is the same for all six programs, and it is an input file that consists of punched cards. Each card included in the file contains one data record that includes from one to eighty characters of information. Each data record represents a purchase made from the reporting firm by a customer. The type of information contained in each record is constant. The types of information, and the card columns in which each appears are shown in Figure 13.

The labels assigned to the fields into which the object program will place the information shown in Figure 13, are as follows:

FIELD	LABEL
Customer Name	NAME
Invoice Date - Month	MONTH
Invoice Date - Day	DAY
Invoice Number	INVNO
Customer Number	CUSTNO
Customer Location - State	STATE
Customer Location - City	CITY
Invoice Amount	INVAMT

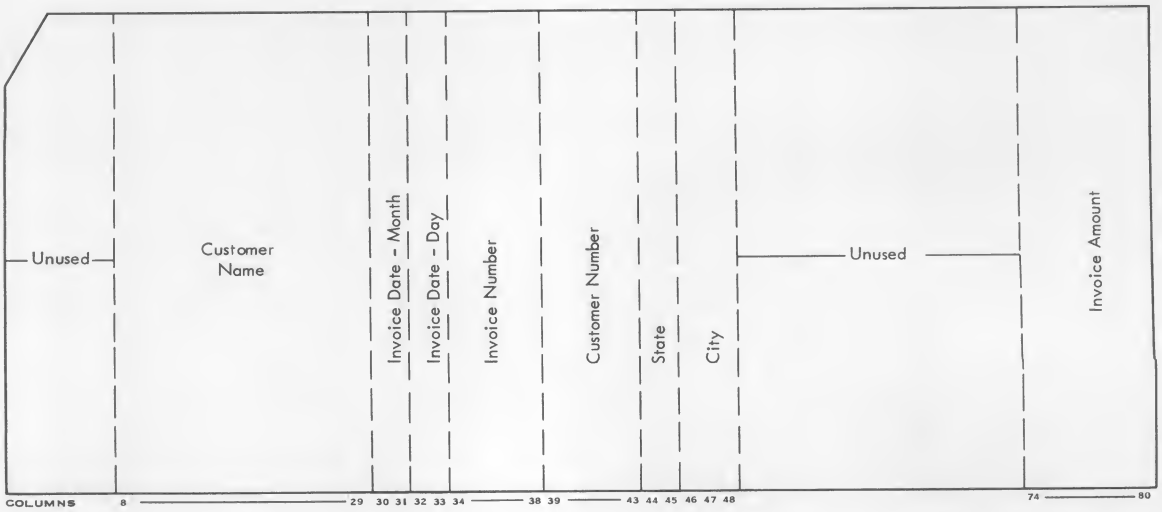

Figure 13. Input File-Card Format

Two figures accompany the discussion of each of the object programs covered in this section. One figure shows the printed report that the object program will produce. The other figure shows the source program coding from which the object program is generated.

Each object program discussed in this section is an expansion or a refinement of the object program that precedes it. As a result, the coding entries required to generate one program are in many instances the same as those needed to generate the preceding object program. When this is the case, the affected coding entries are discussed once, under the program where they are first made. This means, for example, the entries that define the characteristics of the records belonging to the input file common to all object programs, are described only under the discussion of the first object program, although the same entries appear on the source program coding examples that accompany object programs two through six.

Program One (80/80 Listing)

The first object program discussed in this section is capable of reading the records contained in a punched card input file, and writing each of these records, unchanged, on a printer output file. The printed report produced by this program is shown in Figure 14.

```
AMALGAMATED CORP      1110116031071233 61                38925
BROWN WHOLESALE       1228123241131530231                80208
BROWN WHOLESALE       1214995881131530231                26117
FARM IMPLEMENTS       1018109011189747 77                 2763
BLACK OIL             11 8115091853016 67                59295
BLACK OIL             1223122921853016 67                95097
LEATHER BELT CO       11 8115112071536471                33563
LEATHER BELT CO       1217122632071536471                12175
GENERAL MFG CO        11141161529017 6 63                44012
GENERAL MFG CO        11231167629017 6 63                72222
A-B-C DIST CO          911 96892905425 39                64540
A-B-C DIST CO         1111116052905425 39                27169
A-B-C DIST CO         1214122342905425 39                55933
.                      . . .   .   . .  .                .  .
.                      . . .   .   . .  .                .  .
.                      . . .   .   . .  .                .  .
.                      . . .   .   . .  .                .  .
.                      . . .   .   . .  .                .  .
.                      . . .   .   . .  .                .  .
-                     ------  --  ---- --                --  --
8                   29 31 33  38  43 45 48               74  80
PRINT POSITIONS
```

Figure 14. Output - Program One

To produce an object program capable of writing the file shown in Figure 14, the programmer must prepare a source program from the entries made on the coding forms shown in Figure 15.

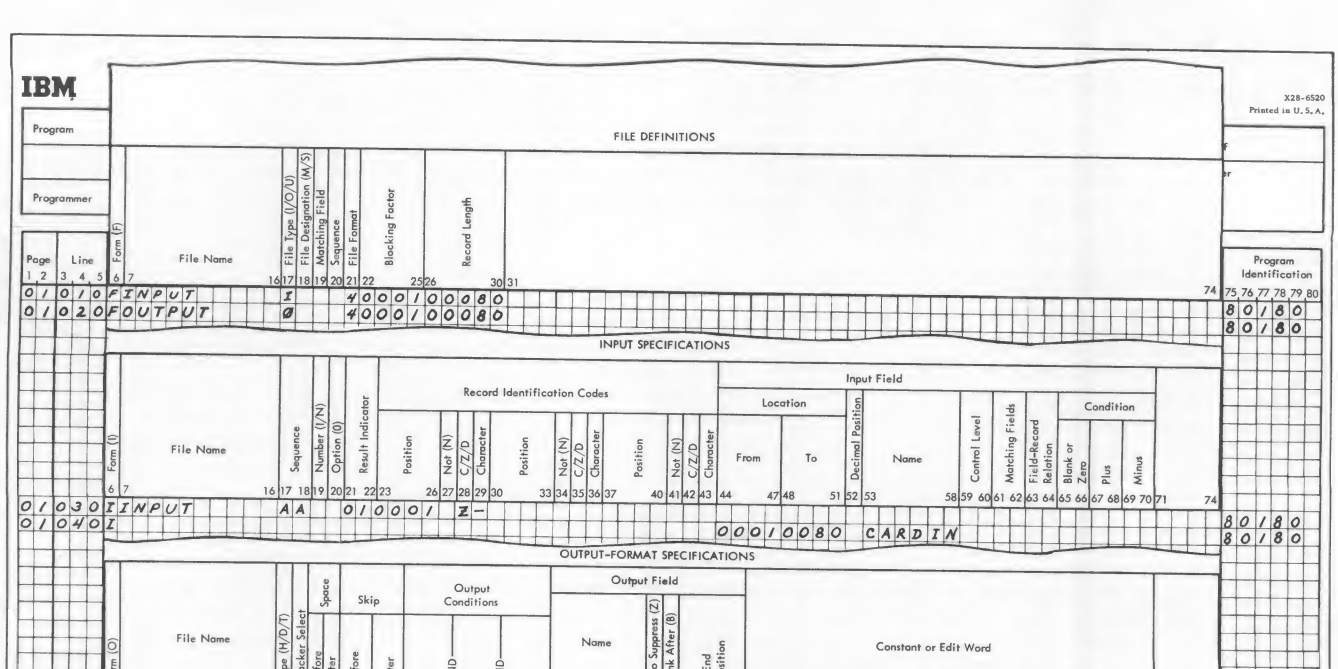

Figure 15. Language Forms - Program One

No Calculation Specifications entries are shown in Figure 15 because no calculations are to be performed by the object program.

NOTE: The column headings shown in Figures 15, 17, 19, 21, 23, and 25 are taken from the RPG coding card. They do not actually appear on the RPG coding form.

File Definitions Entries

The File Definitions entries shown in Figure 15 are discussed in the paragraphs that follow.

File Name (Columns 7-16)

The name of the file defined on line 1 is INPUT. The name of the file defined on line 2 is OUTPUT. Therefore, INPUT is entered, left-justified, in these columns on line 1; OUTPUT, on line 2.

File Type (Column 17)

The file defined on line 1 is an input file. The file defined on line 2 is an output file. As a result, I is entered in this column on line 1; O, on line 2.

File Designation (Column 18)

The file defined on line 1 is the only input file the object program is to reference. The file defined on line 2 is an output file. For this reason, no entry is made in this column on either line 1 or line 2.

Field Sequencing (Columns 19-20)

The object program is not to perform a sequence check on the value of the data contained in a field of each record obtained from the file defined on line 1, and the file defined on line 2 is an output file. Under these circumstances, no entry is made in column 19 or in column 20 on lines 1 or 2.

File Format (Column 21)

The files defined on lines 1 and 2 consist of blocked, fixed-length records. Therefore, the number 4 is entered in this column on lines 1 and 2.

Blocking Factor (Columns 22-25)

There is one record in each block contained in the files defined on lines 1 and 2. As a result, 0001 is entered in these columns on lines 1 and 2.

Record Length (Columns 26-30)

Each record contained in the file defined on line 1 is 80 characters in length. Each record contained in the file defined on line 2 is 80 characters in length. Therefore, 00080 is entered in these columns on both line 1 and line 2.

Input Specifications Entries

The Input Specifications entries shown in Figure 15 are discussed in the paragraphs that follow.

INPUT RECORD DEFINITION ENTRIES

All the data records belonging to the punched card input file have the same characteristics. The input record definition entries made on line 3 define the characteristics of these records.

File Name (Columns 7-16)

The name of the file to which the records defined on the line belong, is INPUT. Therefore, INPUT is entered, left-justified, in these columns.

Sequence (Columns 17-18)

The identifier that is to be assigned to the records whose characteristics are defined on the line, is AA. Therefore, AA is entered in these columns.

Number of Records (Column 19)

The records defined on the line, do not belong to a file that is divided into record groups. As a result, no entry is made in this column.

Option (Column 20)

The records defined on the line are the only records contained in the affected input file, and the affected input file is the only input file that is to be referenced by the object program. Under these conditions, no entry is made in this column.

Result Indicator (Columns 21-22)

The label 01 is to be assigned to the record available result indicator associated with the records defined on the line. Therefore, 01 is entered in columns 21-22.

Record Identification Codes (Columns 23-43)

The presence of a minus sign (-) in the first position (i.e., card column 1) of each data record is the only factor that the object program is to consider in determining whether an input record is to be processed or ignored. In this situation, 0001 is entered in columns 23-26; column 27 is left blank; the letter Z is entered in column 28; a minus sign is entered in column 29; and columns 30-43 are left blank.

INPUT FIELD DEFINITION ENTRIES

Each of the records defined on line 3 contains one field. The *input field* definition entries made on line 4 define this field.

Input Field - From (Columns 44-47)

The first position in the field defined on the line coincides with the first position in each of the records whose characteristics are defined on line 3. Therefore, 0001 is entered in these columns.

Input Field - To (Columns 48-51)

The last position in the field defined on the line is the eightieth position in each of the records defined on the preceding line. Therefore, 0080 is entered in these columns.

Input Field Decimal Position (Column 52)

The field defined on the line does not include an implied decimal point. Therefore, no entry is made in this column.

Input Field Name (Columns 53-58)

The label assigned to the field is CARDIN. Therefore, CARDIN is entered in columns 53-58.

Input Field Control Level (Columns 59-60)

The field defined on the line is not a control field. Therefore, no entry is made in columns 59-60.

Matching Fields (Columns 61-62)

The field defined on the line is not a matching field. Therefore, no entry is made in columns 61-62.

Field-Record Relation (Columns 63-64)

There is only one record group in the input file. As a result, no entry is made in these columns.

Field Condition (Columns 65-70)

No field indicator is to be associated with the field defined on the line. As a result, no entry is made in these columns.

Output-Format Specifications Entries

The Output-Format Specifications entries shown in Figure 15 are discussed in detail in the paragraphs that follow.

OUTPUT RECORD DEFINITION ENTRIES

All the data records that are to be written on the printed report have the same characteristics. The output record definition entries made on line 5 define the characteristics of these records.

File Name (Columns 7-16)

The name of the file to which the records defined on the line belong is OUTPUT. Therefore, OUTPUT is entered, left-justified, in these columns.

Output Record Type (Column 17)

The records defined on the line are detail records. Therefore, the letter D is entered in this column.

Stacker (Column 18)

The output file to which the records defined on the line belong is not a punched card file. Therefore, no entry is made in this column.

Printer Spacing (Columns 19-20)

The print form on which the affected records are to be written is not to be spaced before or after each record is printed. As a result, no entries are made in these columns.

Printer Channel Skipping (Columns 21-24)

The print unit on which the records are to be written, is not to change carriage control tape channels before or after each record is printed. Therefore, no entries are made in these columns.

Output Conditions (Columns 25-33)

The object program is to consider one factor in determining whether a record should be printed. This factor is that the relevant record available result indicator must be ON (i.e., the record available result indicator associated with the input file whose records are defined on line 1). The label assigned this result indicator is 01. Therefore, no entry is made in column 25; 01 is entered in columns 26-27; and no entries are made in columns 28-33.

OUTPUT FIELD DEFINITION ENTRIES

Each of the records whose characteristics are defined on line 5 contains one field. The entries made in the output field definition columns on line 6 define this field.

Output Field Name (Columns 34-39)

The label assigned to the field defined on the line is CARDIN. Therefore, CARDIN is entered, left-justified, in these columns.

Zero Suppression (Column 40)

No zero suppression operation is to be performed on the contents of the field defined on the line before each record containing the field is written by the object program. As a result, no entry is made in this column.

Blank After (Column 41)

The field defined on the line is not to be set to blanks after each record containing the field is printed. Therefore, no entry is made in this column.

Output Field End Position (Columns 42-45)

The last position in the field is the eightieth position in each of the records that contains the field. Therefore, 0080 is entered in these columns.

Constant/Edit Word (Columns 46-70)

No edit operation is to be performed on the contents of the field defined on the line before each record containing the field is printed, and no RPG literal is to be placed in the field at the time the object program is compiled. Under these conditions, no entry is made in these columns.

Program Two (Detail Records)

The second object program, in addition to the functions performed by the first, is capable of reading the same input data records, rearranging the fields contained in each record, and writing the altered records on the printer output file. The printed report produced by this program is shown in Figure 16.

10712	AMALGAMATED CORP	33	61	11603	11	10	38925
11315	BROWN WHOLESALE	30	31	12324	12	28	80208
11315	BROWN WHOLESALE	30	31	99588	12	14	26117
11897	FARM IMPLEMENTS	47	77	10901	10	18	2763
18530	BLACK OIL	16	67	11509	11	8	59295
18530	BLACK OIL	16	67	12292	12	23	95097
20715	LEATHER BELT CO	36	71	11511	11	8	33563
20715	LEATHER BELT CO	36	71	12263	12	17	12175
29017	GENERAL MFG CO	6	63	11615	11	14	44012
29017	GENERAL MFG CO	6	63	11676	11	23	72222
29054	A-B-C DIST CO	25	39	9689	9	11	64540
29054	A-B-C DIST CO	25	39	11605	11	11	27169
29054	A-B-C DIST CO	25	39	12234	12	14	55933
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
18 PRINT POSITIONS	32	59	67	79	88	97	109

Figure 16. Output - Program Two

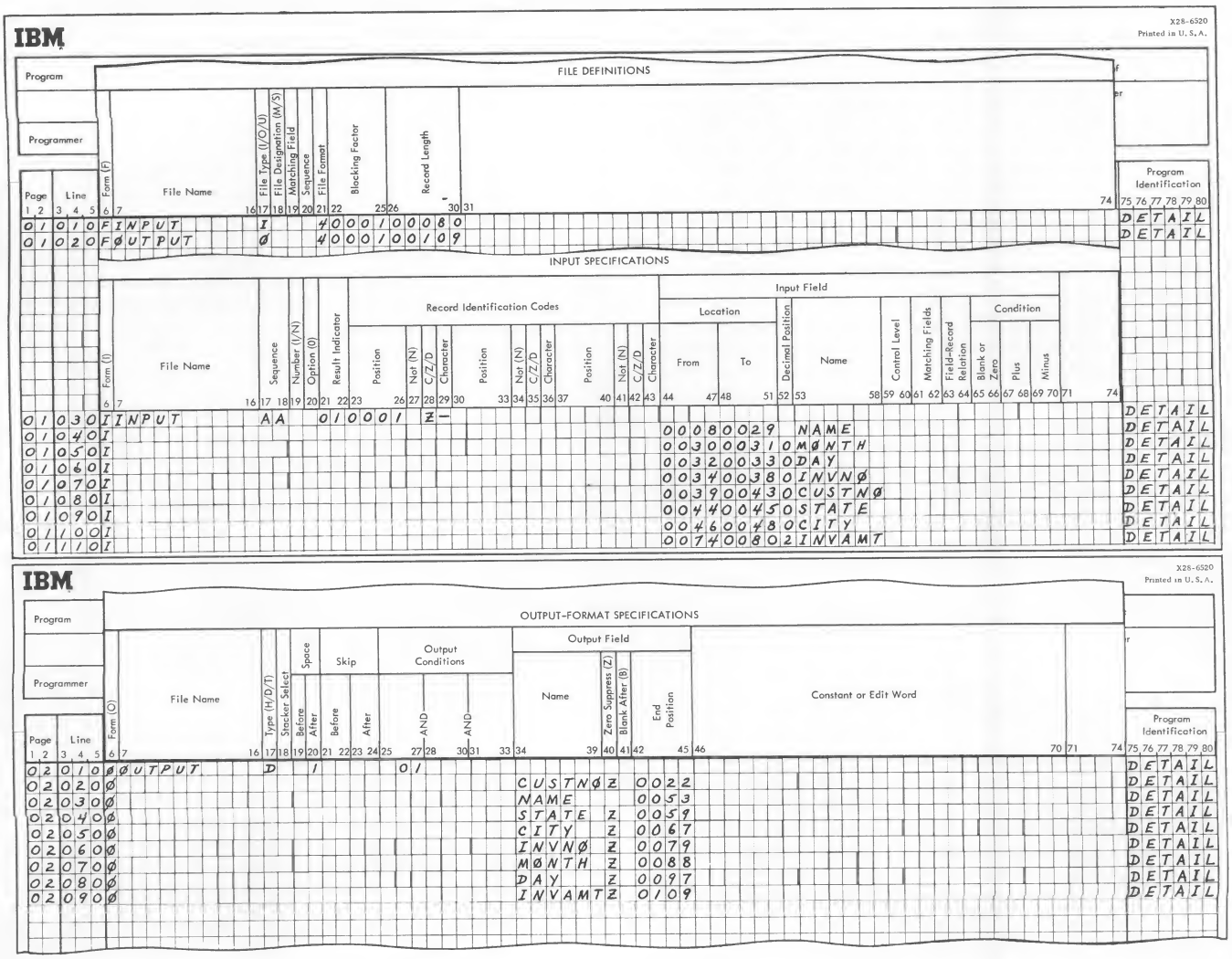

Figure 17. Language Forms - Program Two

To produce an object program capable of printing the report shown in Figure 16, the programmer must prepare a source program from the entries shown in Figure 17.

File Definitions Entries

The File Definitions entries made on page 1, lines 1 and 2 in Figure 17 are the same as those made on lines 1 and 2 of the form in Figure 15, with one exception. Each record contained in the file defined on line 2 is now 109 characters in length. As a result 00109 is entered in columns 26-30 on line 2.

The Input Specifications entries made on page 1, line 3 in Figure 17, are the same as those made on line 3 of the form in Figure 15 and discussed under program one. The entries made on lines 4-11 are discussed in the material that follows.

INPUT FIELD DEFINITION ENTRIES

The records defined on page 1, line 3 each contain eight fields. The input field definition entries made on page 1, lines 4-11 define these fields.

Input Field - From (Columns 44-47)

The first position of the fields defined on page 1, lines 4-11 coincides, respectively, with the 8th, 30th, 32nd, 34th, 39th, 44th, 46th and 74th position of each record that contains the field. As a result, 0008, 0030, 0032, 0034, 0039, 0044, 0046 and 0074 are entered in columns 44-47 on page 1, lines 4-11, respectively.

Input Field - To (Columns 46-51)

The last position in the fields defined on page 1, lines 4-11 coincides, respectively, with the 29th, 31st, 33rd, 38th, 43rd, 45th, 48th and 80th position of each record that contains the field. Therefore, 0029, 0031, 0033, 0038, 0043, 0045, 0048 and 0080 are entered in columns 46-51 on page 1, lines 4-11, respectively.

Input Field Decimal Position (Column 52)

The data in the field defined on page 1, line 4 does not contain an implied decimal point. As a result, no entry is made in column 52 on this line.

The fields defined on page 1, lines 5-10 contain numeric information with no positions of data to the right of the implied decimal point contained in the data. Therefore, the number 0 is entered in column 52 on each of these lines.

The field defined on page 1, line 11 contains two positions of data that are to the right of the position occupied by the implied decimal point contained in the data. Therefore, the number 2 is entered on the line in column 52.

Input Field Name (Columns 53-58)

The labels assigned to the fields defined on page 1, lines 4-11 are, respectively, NAME, MONTH, INVNO, CUSTNO, STATE, CITY, and INVAMT. As a result, each of these labels is entered in columns 53-58 on the appropriate line.

Input Field Control Level (Columns 59-60)

None of the fields defined on page 1, lines 4-11 are control fields. For this reason, no entry is made in columns 59-60 on any of these lines.

Matching Fields (Columns 61-62)

None of the fields defined on page 1, lines 4-11 have been designated matching fields. As a result, no entry is made in columns 61-62 on these lines.

Field-Record Relation (Column 63-64)

There is only one record group in the input file. Therefore, no entry is made in these columns.

Input Field Condition (Columns 65-70)

No field indicator is to be assigned to the fields defined on these lines. Therefore, no entry is made in these columns.

Output-Format Specifications Entries

The Output-Format Specifications entries made on page 2, line 1 of the form shown in Figure 17 are the same as those made on line 5 of the form in Figure 15 and discussed under program one, with the exception of the printer spacing entry. The number 1 is entered in column 20 to specify that the print form is to be spaced 1 line after printing each record. The entries made on page 2, lines 2-9 of the form shown in Figure 17 are discussed in the paragraphs that follow.

OUTPUT FIELD DEFINITION ENTRIES

Each of the records whose characteristics are defined on page 2, line 1 contains eight data fields. The field definition entries made on page 2, lines 2-9 define these fields.

Output Field Name (Columns 34-39)

The labels of the fields defined on page 2, lines 2-9, are, respectively, CUSTNO, NAME, STATE, CITY, INVNO, MONTH, DAY, and INVAMT. As a result, these labels are entered in columns 34-39 on the appropriate lines.

Zero Suppression (Column 40)

Zeros are to be eliminated from the high-order positions of the fields defined on page 2, line 2 and page 2, lines 4-9 before each record that contains these fields is written on the printed report. In this situation, the letter Z is entered in column 40 on these lines.

High-order zeros are not to be eliminated from the field defined on page 2, line 3. Therefore, no entry is made in column 40 on the line.

Blank After (Column 41)

None of the fields defined on page 2, lines 2-9 are to be set to blanks each time a record containing the fields is output by the object program. For this reason, no entry is made in column 41 on these lines.

Output Field End Position (Columns 42-45)

The last position in the fields defined on page 2, lines 2-9 coincides, respectively, with the 22nd, 53rd, 59th, 67th, 79th, 88th, 97th, and 109th position in each record that contains the field. As a result, 0022, 0053, 0059, 0067, 0079, 0088, 0097, and 0109 are entered in columns 42-45 on page 2, lines 2-9, respectively.

Constant/Edit Word (Columns 46-70)

No edit operation is to be performed on the contents of the fields defined on page 2, lines 2-9 before each record containing the fields is written by the object program, and no RPG literal is to be placed in any of these fields at the time that the object program is compiled. Under these conditions, no entry is made in columns 46-70 on any of these lines.

Program Three (Final Total)

The third object program, in addition to the functions performed by the second, is capable of accummulating data for a total record that represents the total amount owed the reporting firm by all its customers (i.e., a final total). In addition, it is capable of printing this record on the printer output file, once the last input record has been obtained and the last detail record has been printed. The printed report produced by this program is shown in Figure 18.

```
10712        AMALGAMATED CORP      33      61      11603       11      10       38925
11315        BROWN WHOLESALE       30      231     12324       12      28       80208
11315        BROWN WHOLESALE       30      231     99588       12      14       26117
11897        FARM IMPLEMENTS       47      77      10901       10      18        2763
18530        BLACK OIL             16      67      11509       11      8        59295
18530        BLACK OIL             16      67      12292       12      23       95097
20715        LEATHER BELT CO       36      471     11511       11      8        33563
20715        LEATHER BELT CO       36      471     12263       12      17       12175
29017        GENERAL MFG CO         6      63      11615       11      14       44012
29017        GENERAL MFG CO         6      63      11676       11      23       72222
29054        A-B-C DIST CO         25      39       9689        9      11       64540
29054        A-B-C DIST CO         25      39      11605       11      11       27169
29054        A-B-C DIST CO         25      39      12234       12      14       55933
.            .                      .       .       .           .       .
.            .                      .       .       .           .       .    1752423**
.            .                      .       .       .           .       .       . .
.            .                      .       .       .           .       .       . .
.            .                      .       .       .           .       .       . .
18           32                    59      67      79          88      97      109  111
PRINT
POSITIONS
```

Figure 18. Output - Program Three

To produce an object program capable of writing the output file shown in Figure 18, the programmer must prepare a source program from the entries shown in Figure 19.

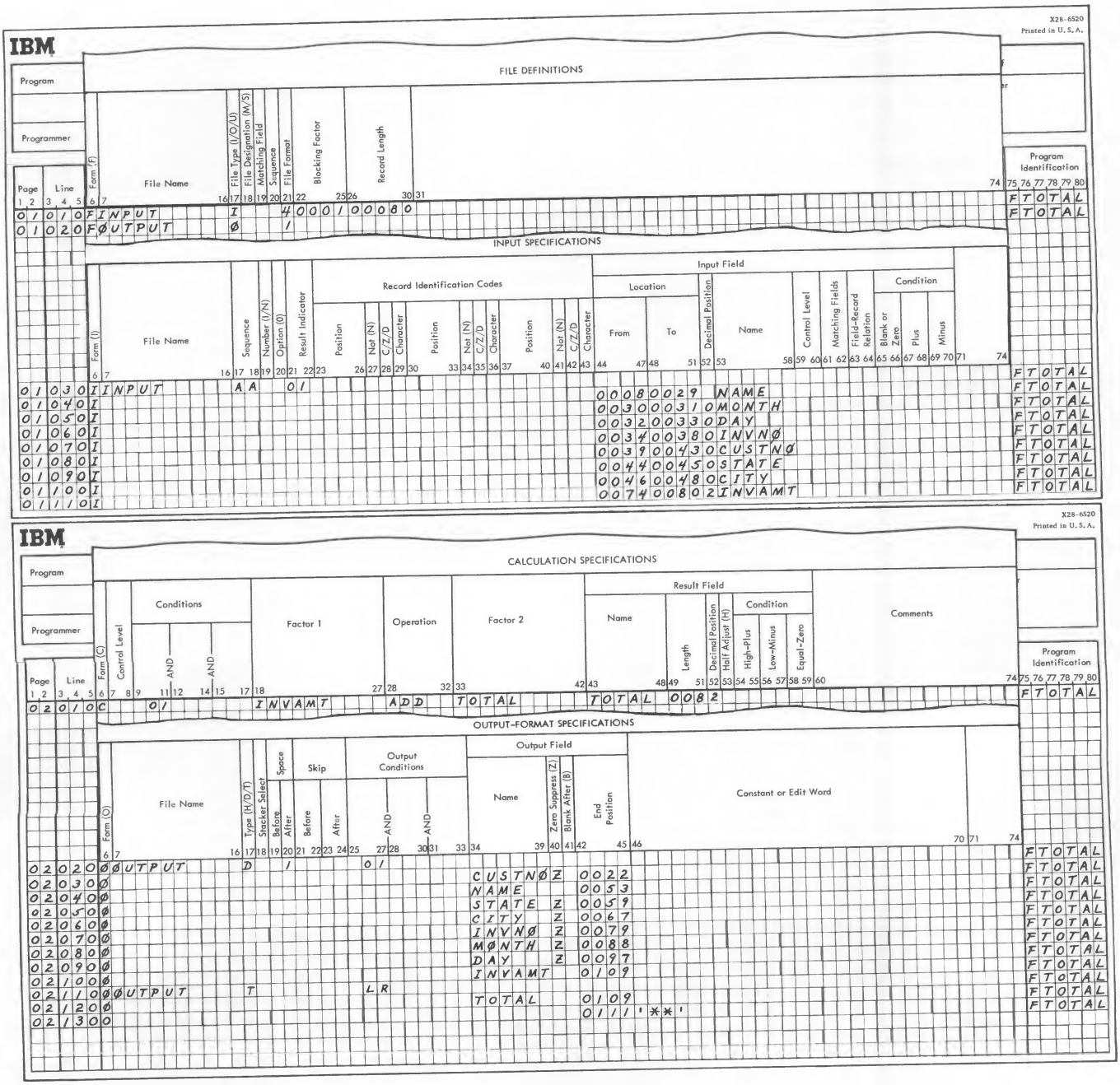

Figure 19. Language Forms - Program Three

Several of the File Definitions entries made on page 1, line 2; the Calculation Specifications entries made on page 2, line 1; and the Output-Format Specifications entries made on page 2, lines 11-13, are the only entries shown in Figure 19 that do not appear in the coding example (i.e., Figure 17) accompanying the discussion of the second object program. They are also the only entries that have not been explained in the discussions of program one or two. They are explained in the material that follows.

File Definitions Entries

The File Definitions entries in Figure 19 differ somewhat from those made in Figure 17. The new entries are discussed below.

File Format (Column 21)

The file defined on page 1, line 2 now consists of unblocked, variable-length records. Therefore, the number 1 is entered in this column on page 1, line 2.

Blocking Factor (Columns 22-25)

No entry is made in these columns on page 1, line 2 because the file defined on the line now consists of unblocked, variable-length records.

Record Length (Columns 26-30)

No entry is made in these columns on page 1, line 2 because the file defined on the line now consists of unblocked, variable-length records.

Calculation Specifications Entries

The Calculation Specifications entries made on page 2, line 1 are discussed in the material that follows.

Control Level (Columns 7-8)

Execution of the operation defined on the line cannot be made conditional on the occurrence of a control break because none of the fields contained in the input data records are control fields. As a result, no entry is made in these columns.

Conditions (Columns 9-17)

The execution of the operation defined on the line is dependent on the record available result indicator labeled 01 being ON. Therefore, the entry 01 is made in columns 10-11.

Factor 1 (Columns 18-27)

The label of the field whose contents are to be added to the contents of a field in which a total is being accumulated, is INVAMT. Therefore, the label INVAMT is entered in columns 18-27.

Operation Code (Columns 28-32)

The calculation defined on the line is an add operation. Therefore, the operation code ADD is entered in these columns.

Factor 2 (Columns 33-42)

The field to which the contents of INVAMT are to be added, is labeled TOTAL. Therefore, the label TOTAL is entered in these columns.

Result Field Name (Columns 43-48)

The results of the add operation defined on the line are to be stored in the field labeled TOTAL. Therefore, the label TOTAL is entered in these columns.

Result Field Length (Columns 49-51)

The length of the field labeled TOTAL, expressed as a three-digit number, is 008. Therefore, 008 is entered in these columns.

Result Field Decimal Position (Column 52)

There are two positions to the right of the implied decimal point in the data contained in the field labeled TOTAL. Therefore, the number 2 is entered in this column.

Result Field Half Adjust (Column 53)

No half adjust operation is to be performed on the contents of the field labeled TOTAL. Therefore, no entry is made in this column.

Condition (Columns 54-59)

No result indicators are to be turned ON as a result of the operation defined on the line. Therefore, no entries are made in these columns.

Output-Format Specifications Entries

The Output-Format Specifications entries made on page 2, lines 11-13 are discussed in detail in the paragraphs that follow.

OUTPUT RECORD DEFINITION ENTRIES

The printed report shown in Figure 18, contains a total record. The cnaracteristics of the record group to which this record belongs are defined in the output record definition columns on page 2, line 11.

File Name (Columns 7-16)

The name of the file to which the records defined on the line belong, is OUTPUT. Therefore, OUTPUT is entered in columns 7-16.

Output Record Type (Column 17)

The records whose characteristics are defined on page 2, line 11 are total records. Therefore, the letter T is entered in column 17.

Stacker (Column 18)

The file to which the records defined on the line belong is not a punched card file. For this reason, no entry is made in this column.

Printer Spacing (Columns 19-20)

The standard print form is not to be spaced before or after each record defined on the line is printed. As a result no entry is made in column 19 and no entry is made in column 20.

Printer Channel Skipping (Columns 21-24)

The printer is not to change carriage control tape channels before or after printing each of the affected records. Under these conditions, no entry is made in columns 21-22 and no entry is made in columns 23-24.

Output Conditions (Columns 25-33)

The record defined on page 2, line 11 is to be printed whenever the object program sets the LR (last record) special indicator ON. Therefore, the label LR is entered in columns 26-27.

OUTPUT FIELD DEFINITION ENTRIES

The total record defined on page 2, line 11 contains two fields. The output field definition entries on page 2, lines 12 and 13 are used to define these fields.

Output Field Name (Columns 34-39)

The label of the field defined on page 2, line 12 is TOTAL. Therefore, TOTAL is entered in columns 34-43 on that line. The field defined on page 2, line 13 is to contain a literal. Therefore, no entry is made in columns 34-39 on that line.

Zero Suppression (Column 40)

A zero suppression operation is to be performed on the contents of the field defined on page 2, line 12 before each record containing the field is printed by the object program. As a result, the letter Z is entered in column 40 of that line.

No zero suppression operation is to be performed on the contents of the field defined on page 2, line 13. Therefore, no entry is made in column 44 of the line.

Blank After (Column 41)

Neither the field defined on page 2, line 12, nor the field defined on page 2, line 13 are to be set to blanks after each record containing the fields is output by the object program. Therefore, no entry is made in column 41 on page 2 on either line 12 or on line 13.

Output Field End Position (Columns 42-45)

The last position in the field defined on page 2, line 12 coincides with the 109th position in each record that contains the field. For this reason, 0109 is entered in columns 42-45.

The last position of the field defined on page 2, line 13 coincides with the 111th position of the record that contains the field. Therefore, 0111 is entered in columns 42-45.

Constant/Edit Word (Columns 46-70)

No editing is to be performed on the contents of the field defined on page 2, line 12 before each record containing the field is printed by the object program, and no RPG literal is to be placed in the field at the time the object program is compiled. Therefore, no entry is made in columns 46-70 on page 2, line 12.

The literal ** (two asterisks) is to be placed in the field defined on page 2, line 13 at the time the object program is being compiled. As a result, the entry '**' is placed in columns 46-70, on line 13.

Program Four (Editing)

The fourth object program, in addition to the functions performed by the third, is capable of causing an edit operation to be performed on the contents of a field in each detail record, and on the contents of each total record that is to be printed on the report. The printed report produced by this program is shown in Figure 20.

```
10712          AMALGAMATED CORP          33    61     11603     11     10 $     389.25
11315          BROWN WHOLESALE           30   231     12324     12     28 $     802.08
11315          BROWN WHOLESALE           30   231     99588     12     14 $     261.17
11897          FARM IMPLEMENTS           47    77     10901     10     18 $      27.63
18530          BLACK OIL                 16    67     11509     11      8 $     592.95
18530          BLACK OIL                 16    67     12292     12     23 $     950.97
20715          LEATHER BELT CO           36   471     11511     11      8 $     335.63
20715          LEATHER BELT CO           36   471     12263     12     17 $     121.75
29017          GENERAL MFG CO             6    63     11615     11     14 $     440.12
29017          GENERAL MFG CO             6    63     11676     11     23 $     722.22
29054          A-B-C DIST CO             25    39      9689      9     11 $     645.40
29054          A-B-C DIST CO             25    39     11605     11     11 $     271.69
29054          A-B-C DIST CO             25    39     12234     12     14 $     559.33
.              .                          .     .         .      .      .
.              .                          .     .         .      .      . $ 17,524.23**
.              .                          .     .         .      .      .          . .
.              .                          .     .         .      .      .          . .
.              .                          .     .         .      .      .          . .
18             32                        59    67        79     88     97        109 111
PRINT
POSITIONS
```

Figure 20. Output - Program Four

To produce an object program capable of writing the output file shown in Figure 20, the programmer must prepare a source program from the entries shown in Figure 21.

The Output-Format Specifications entries made in columns 50-74 on page 2, lines 10 and 12 are the only entries shown in Figure 21 that do not appear in the coding example (i.e., Figure 19) included in the discussion of the third object program. They are also the only entries that have not been explained in the discussions accompanying programs one through three. They are explained in the paragraphs that follow.

Constant/Edit Word (Columns 50-74)

The contents of the fields defined on page 2, lines 10 and 12 are to be edited before each record containing these fields is printed by the object program. For this reason, the edit word (i.e., '$bbb,bb0.bb') that is to be referenced by the edit operations, is placed in columns 50-74 on page 2, on both line 10 and line 12. (The zero in the units position of the edit word will cause zeros to be suppressed in the relevant fields up to and including the units position. For this reason, no entry is made in column 40 on page 2 on lines 10 and 12.)

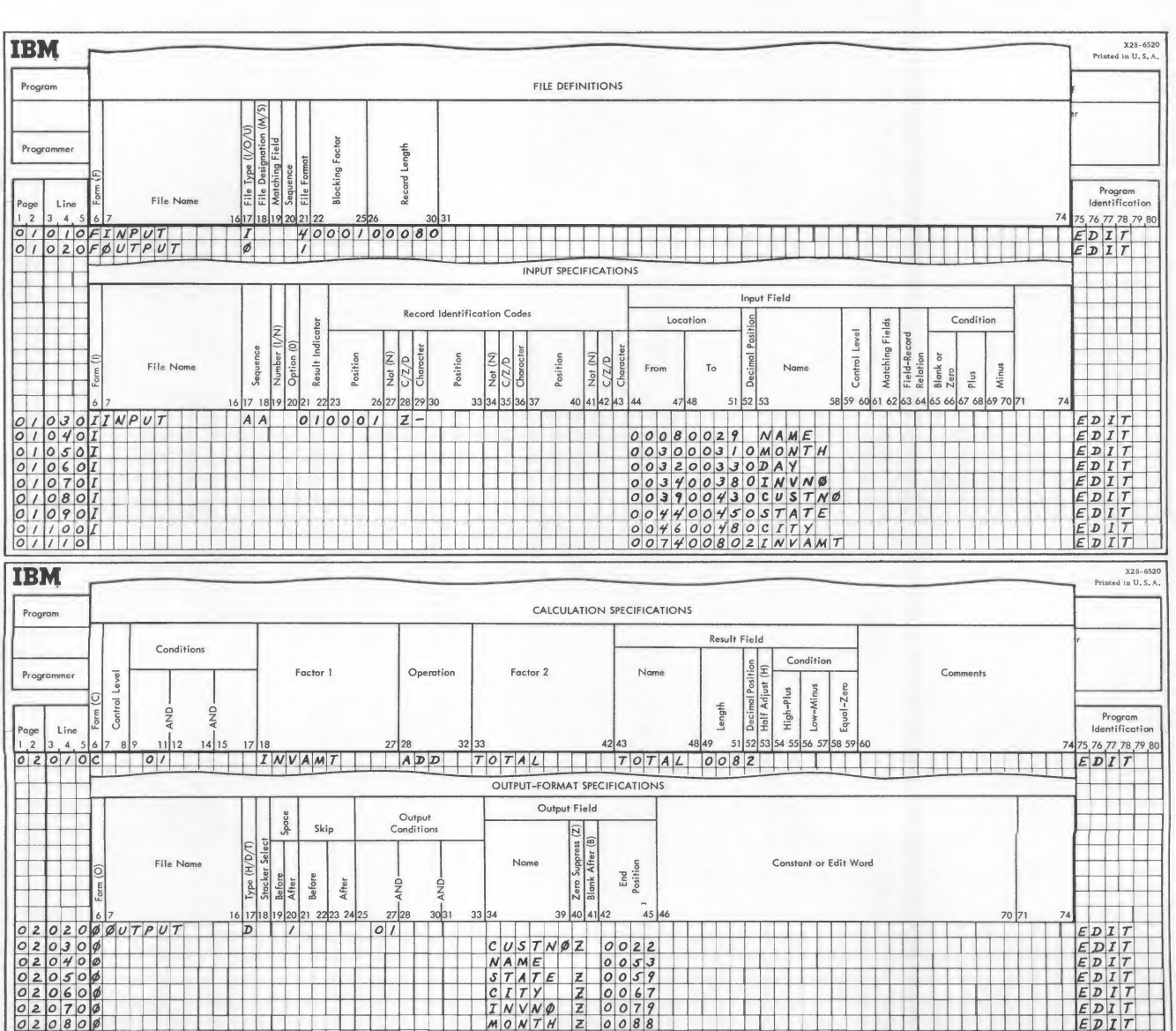

Figure 21. Language Forms - Program Four

Program Five (Minor Totals)

The fifth object program, in addition to the functions performed by the fourth, is capable of accumulating data for total records that represent the total amount owed the reporting firm by one customer (i.e., minor totals). In addition, it is capable of writing one of these records on the printer output file whenever the contents of the control field CUSTNO change (i.e., a record is obtained that represents a purchase made by another customer). The printed report produced by this program is shown in Figure 22.

```
10712          AMALGAMATED CORP      33      61      11603       11      10 $     389.25
                                                                                $     389.25*
11315          BROWN WHOLESALE       30     231      12324       12      28 $     802.08
11315          BROWN WHOLESALE       30     231      99588       12      14 $     261.17
                                                                                $   1,063.25*
11897          FARM IMPLEMENTS       47      77      10901       10      18 $      27.63
                                                                                $      27.63*
18530          BLACK OIL             16      67      11509       11       8 $     592.95
18530          BLACK OIL             16      67      12292       12      23 $     950.97
                                                                                $   1,543.92*
20715          LEATHER BELT CO       36     471      11511       11       8 $     335.63
20715          LEATHER BELT CO       36     471      12263       12      17 $     121.75
                                                                                $     457.38*
29017          GENERAL MFG CO         6      63      11615       11      14 $     440.12
29017          GENERAL MFG CO         6      63      11676       11      23 $     722.22
                                                                                $   1,162.34*
29054          A-B-C DIST CO         25      39       9689        9      11 $     645.40
29054          A-B-C DIST CO         25      39      11605       11      11 $     271.69
29054          A-B-C DIST CO         25      39      12234       12      14 $     559.33
.              .                      .       .          .        .       .
.              .                      .       .          .        .       .  $   1,476.42*
.              .                      .       .          .        .       .
.              .                      .       .          .                  $  17,524.23**
.              .                      .       .          .        .       .          .  .
18             32                    59      67         79       88      97        109 111
PRINT
POSITIONS
```

Figure 22. Output - Program Five

To produce an object program capable of writing the output file shown in Figure 22, the programmer must prepare a source program from the entries shown in Figure 23.

The only entries shown in Figure 23 that do not appear in the coding example (i.e., Figure 21) included in the discussion of the fourth object program are: (a) one Input Specifications entry made on page 1, line 8, (b) the Calculation Specifications entries made on page 2, line 2, and (c) the Output-Format Specifications entries made on page 2, lines 15-17. They are also the only entries that have not been explained in the discussions accompanying programs one through four. They are explained in the material that follows.

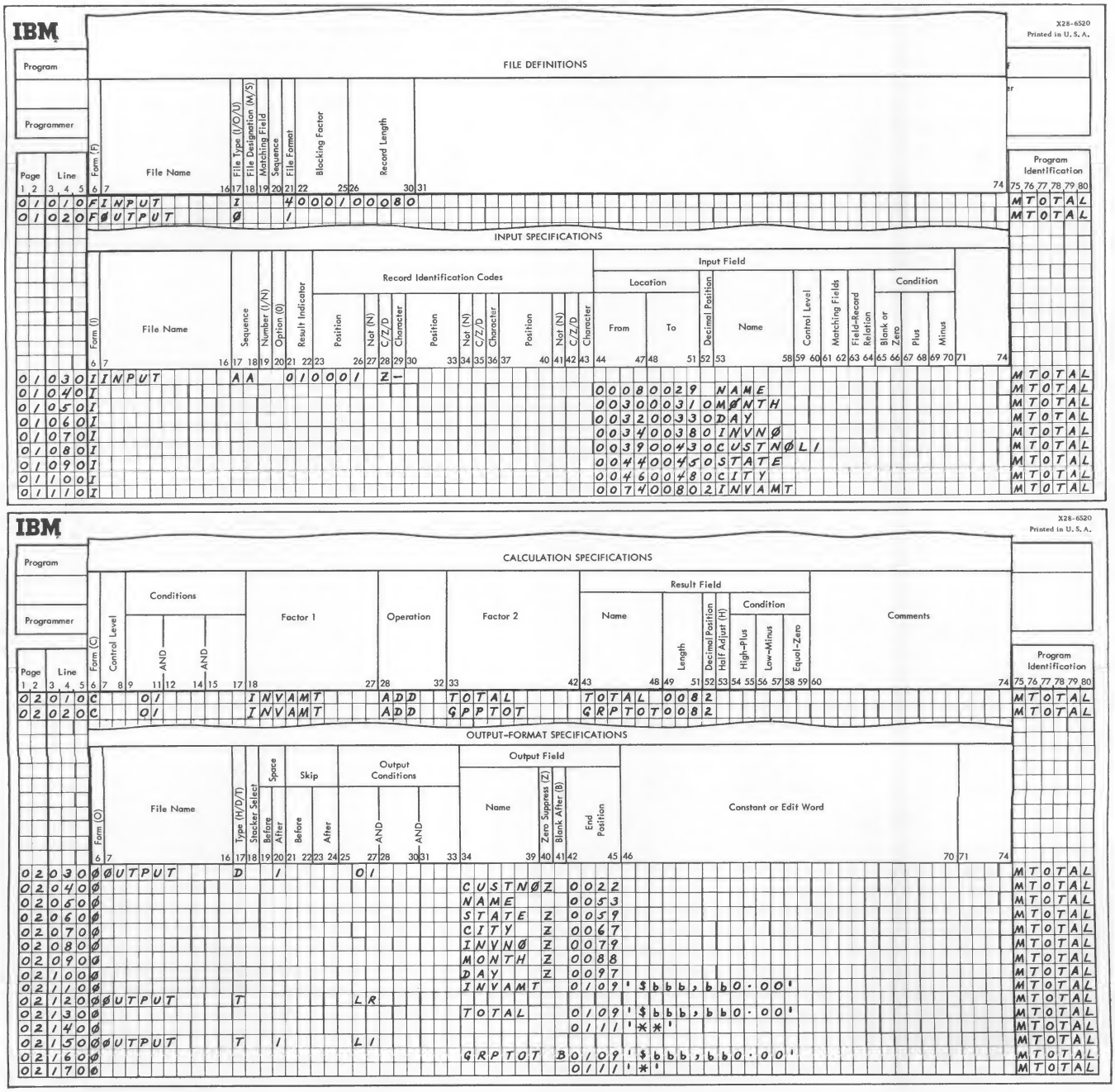

Figure 23. Language Forms - Program Five

Input Specifications Entries

The Input Specifications entry made on page 1, line 8 is discussed below.

Control Level (Columns 59-60)

The field defined on page 1, line 8 is now a control field. The control level that is to be assigned to this field is L1. Therefore, the code L1 is entered on page 1, line 8 in columns 59-60.

Calculation Specifications Entries

The Calculation Specifications entries made on page 2, line 2 are discussed in the material that follows.

Control Level (Columns 7-8)

The execution of the operation defined on page 2, line 2 is not dependent on a control break. For this reason, no entry is made in these columns.

Conditions (Columns 9-17)

The record available result indicator labeled 01 must be ON for the operation defined on page 2, line 2 to be executed. Therefore, the entry 01 is made in these columns.

Factor 1 (Columns 18-27)

The contents of the field labeled INVAMT are to be added to the contents of a field in which a minor total is being accumulated. Therefore, the label INVAMT is entered in columns 18-27 on page 2, line 2.

Operation Code (Columns 28-32)

The calculation defined on the line is an add operation. As a result, the operation code ADD is entered in columns 28-32.

Factor 2 (Columns 33-42)

The field to which the contents of INVAMT are to be added, is labeled GRPTOT. Therefore, the label GRPTOT is entered in columns 33-42 on page 2, line 2.

Result Field Name (Columns 43-48)

The results of the add operation defined on page 2, line 2 are to be stored in the field labeled GRPTOT. For this reason, the label GRPTOT is entered in columns 43-48 of the line.

Result Field Length (Columns 49-51)

The field labeled GRPTOT is 8 characters in length. Therefore, 008 is entered in columns 49-51 of the line.

Result Field Decimal Position (Column 52)

The data placed in the result field labeled GRPTOT contains two positions that are to the right of the implied position of the decimal point contained in the data. As a result, the number 2 is entered in column 52 of the line.

Result Field Half Adjust (Column 53)

No half adjust operation is to be performed on the contents of the field labeled GRPTOT. Therefore, no entry is made in column 53 of the line.

Result Field Condition (Columns 54-59)

No result indicators are to be turned ON as a result of the operation defined on the line. Therefore, no entries are made in columns 54-59 of the line.

Output-Format Specifications Entries

The Output-Format Specifications entries made on page 2, lines 15-17 are discussed in the paragraphs that follow.

OUTPUT RECORD DEFINITION ENTRIES

The printed report shown in Figure 22 contains an additional group of total records. The characteristics of the records belonging to this additional group are defined on page 2, line 15.

File Name (Columns 7-16)

OUTPUT is the name of the file to which the records defined on the line belong. As a result, OUTPUT is entered, left-justified, in these columns.

Output Record Type (Column 17)

The records belonging to the record group defined on line 15 are total records. For this reason, the letter T is entered in column 17 of the line.

Stacker (Column 18)

The output file to which the records defined on the line belong is not a punched card file. Therefore, no entry is made in this column.

Printer Spacing (Columns 19-20)

The print form on which the relevant output file is written is not to be spaced before a total record belonging to the record group defined on Line 15 is printed. Therefore, no entry is made in column 19 on the line. The print form is to be spaced one line after a record belonging to the group defined on the line is printed. In this situation, the number 1 is entered on the line in column 20.

Printer Channel Skipping (Columns 21-24)

The printer is not to skip carriage control tape channels before or after printing a total record belonging to the group defined on page 2, line 15. Therefore, no entries are made on the line in columns 21-24.

Output Conditions (Columns 25-33)

A total record belonging to the group defined on page 2, line 15 is to be printed whenever a control break occurs in the control field labeled CUSTNO. (A control break is said to have occurred in the control field labeled CUSTNO whenever the object program obtains an input record that contains a customer number that is different from the customer number contained in the last input record processed by the object program.) To indicate that this is the case, L1 (i.e., the control level assigned to the field labeled CUSTNO) is entered in columns 26-27.

OUTPUT FIELD DEFINITION COLUMNS

The fields contained in each total record belonging to the record group defined on page 2, line 15, are defined on page 2, lines 16 and 17.

Output Field Name (Columns 34-39)

The label of the field defined on page 2, line 16 is GRPTOT. Therefore, GRPTOT is entered in columns 34-39.

The field defined on page 2, line 17 is to contain a literal. For this reason, no entry is made in columns 34-39 on the line.

Zero Suppression (Column 40)

No zero suppression operation is to be performed on the contents of the fields defined on page 2, lines 16-17. Therefore, no entry is made on these lines in column 40.

Blank After (Column 41)

The field defined on page 2, line 16 is to be set to zero after each total record containing the field is written by the object program. In this situation, the letter B is entered on the line in column 41.

The field defined on page 2, line 17 is not to be set to blanks after each record containing the field is printed. Therefore, no entry is made on the line in column 41.

Output Field End Position (Columns 42-45)

The last position in the field defined on page 2, line 16 coincides with the 109th position of the record that contains the field; the last position in the field defined on page 2, line 17, with the 111th position. Therefore, 0109 is entered in columns 42-45 on page 2, line 16; 0111, on page 2, line 17.

Constant/Edit Word (Columns 46-70)

The contents of the field defined on page 2, line 16, is to be edited before each record that contains the field is printed by the object program. Under these conditions, the edit word that is to be referenced by the edit operation is placed in columns 46-70 of the line.

The literal * (asterisk) is to be placed in the field defined on page 2, line 17 at the time the object program is compiled. Therefore, the entry '*' is placed on the line in columns 46-70.

Program Six (Header Records)

The sixth object program, in addition to the functions performed by the fifth object program, is capable of writing certain specified header records at the beginning of each page of the print form on which the output file is written. The printed report produced by this program is shown in Figure 24.

To produce an object program capable of writing the report shown in Figure 24, the programmer must prepare a source program from the entries shown in Figure 25.

The Output-Format Specifications entries made on page 2, lines 3-16 and on page 3, lines 1-3 are the only entries shown in Figure 25 that do not appear in the coding example (i.e., Figure 23) included in the discussion of the fifth object program. They are also the only entries that have not been explained in the discussions accompanying programs one through five. They are explained in the paragraphs that follow.

OUTPUT RECORD DEFINITION ENTRIES

The printed report shown in Figure 24 contains header records that belong to three distinct record groups. The characteristics of the records belonging to each of these groups are defined on page 2, lines 3-4, 8-9, and 14-15 shown in Figure 25.

File Name (Columns 7-16)

OUTPUT is the name of the file to which the records defined on the line belong. Therefore, OUTPUT is entered, left-justified, in these columns on page 2, lines 3, 8, and 14.

A C C O U N T S R E C E I V A B L E R E G I S T E R

CUSTOMER NUMBER	CUSTOMER NAME	LOCATION STATE	LOCATION CITY	INVOICE NUMBER	INVOICE DATE MO	INVOICE DATE DAY		INVOICE AMOUNT
10712	AMALGAMATED CORP	33	61	11603	11	10	$	389.25
							$	389.25*
11315	BROWN WHOLESALE	30	231	12324	12	28	$	802.08
11315	BROWN WHOLESALE	30	231	99588	12	14	$	261.17
							$	1,063.25*
11897	FARM IMPLEMENTS	47	77	10901	10	18	$	27.63
							$	27.63*
18530	BLACK OIL	16	67	11509	11	8	$	592.95
18530	BLACK OIL	16	67	12292	12	23	$	950.97
							$	1,543.92*
20715	LEATHER BELT CO	36	471	11511	11	8	$	335.63
20715	LEATHER BELT CO	36	471	12263	12	17	$	121.75
							$	457.38*
29017	GENERAL MFG CO	6	63	11615	11	14	$	440.12
29017	GENERAL MFG CO	6	63	11676	11	23	$	722.22
							$	1,162.34*
29054	A-B-C DIST CO	25	39	9689	9	11	$	645.40
29054	A-B-C DIST CO	25	39	11605	11	11	$	271.69
29054	A-B-C DIST CO	25	39	12234	12	14	$	559.33
.	.	.	.	.	.	.		
.	.	.	.	.	.	.	$	1,476.42*
.	.	.	.	.	.	.		
.	.	.	.	.	.	.	$	17,524.23**
.	.	.	.	.	.	.		. .
.	.	.	.	.	.	.		. .
18 PRINT POSITIONS	32	59	67	79	88	97		109 111

Figure 24. Output - Program Six

Output Record Type (Column 17)

The records belonging to the record groups defined on page 2, lines 3-4, 8-9, and 14-15 are all header records. Therefore, the letter H is entered in column 17 on page 2, lines 3, 8, and 14.

Stacker (Column 18)

The records defined on the line belong to a printer file. Therefore, no entry is made in this column.

Printer Spacing (Columns 19-20)

The print form on which the output file is written is not to be spaced before a header record belonging to any of the three groups defined on page 2, lines 3-4, 8-9, and 14-15 is printed. Therefore, no entry is made in column 19 on page 2, lines 3, 8, and 14.

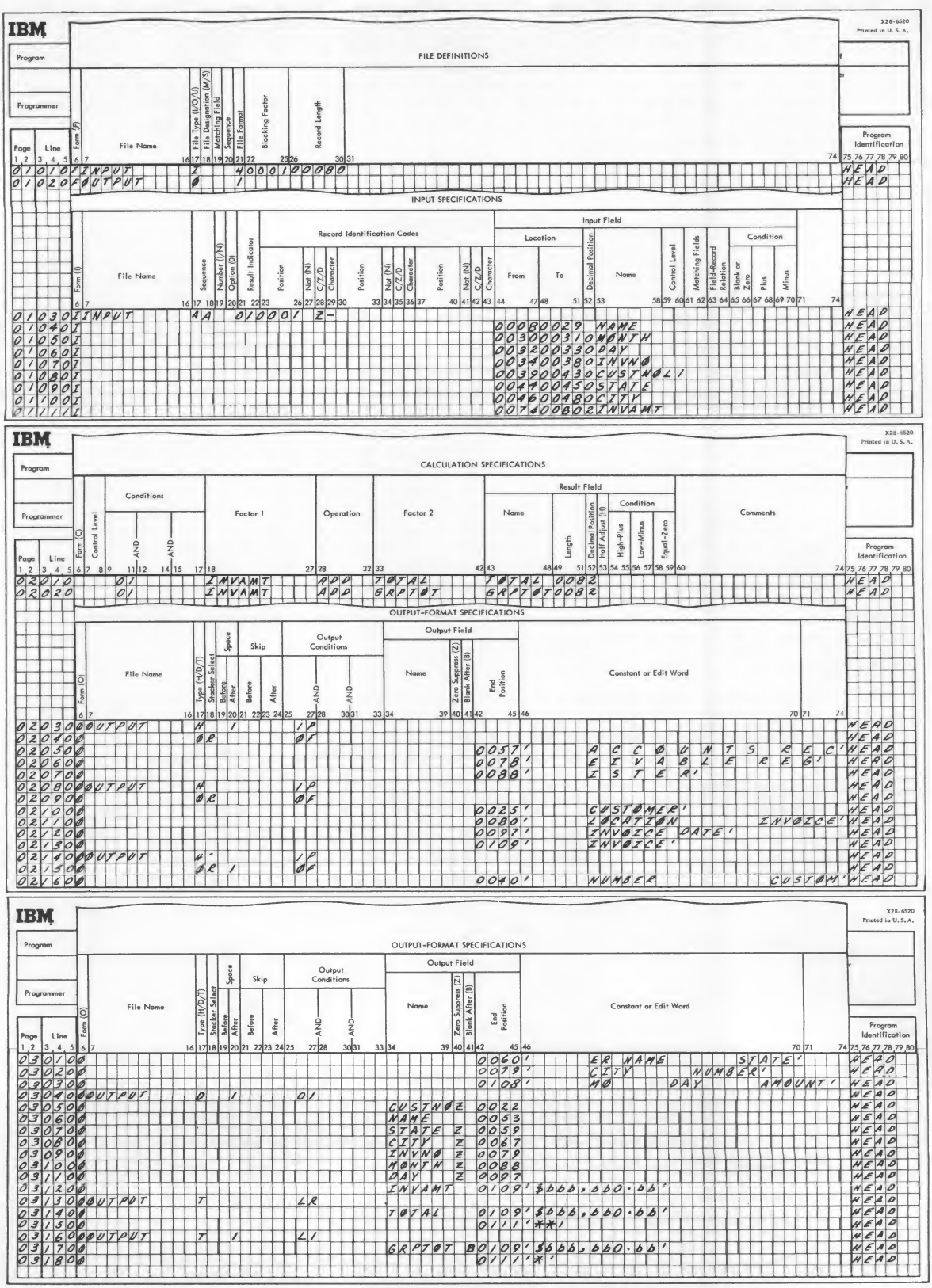

Figure 25. Language Forms - Program Six

The print form on which the output file is written, is to be spaced one line after a header record belonging to the group defined on page 2, lines 3-4 is printed; no lines after a header record belonging to the group defined on page 2, lines 8-9 is printed; and one line after a header record belonging to the group defined on page 2, lines 14-15 is printed. As a result, the number 1 is entered in column 20 on page 2, line 3; no entry is made in column 20 on page 2, line 8; and the number 1 is entered in column 20 on page 2, line 14.

Printer Channel Skipping (Columns 21-24)

The printer is not to skip carriage control tape channels before or after a header record from the groups defined on page 2, lines 3-4, 8-9, and 14-15 is printed. Under these conditions no entry is made in columns 21-24 on these lines.

Output Conditions (Columns 25-33)

A header record belonging to each of the record groups under discussion is to be written whenever the first page special indicator, or the page overflow special indicator is ON. As a result, no entry is made in column 25 on page 2, lines 3-4, 8-9, and 14-15; the label of the first page special indicator is entered in columns 26-27 on page 2, lines 3, 8, and 14; the label of the page overflow special indicator is entered in the same columns on page 2, lines 4, 9, and 15; and the letters OR are entered in columns 17-18 on page 2, lines 4, 9, and 15.

OUTPUT FIELD DEFINITION ENTRIES

The fields contained in each header record belonging to the group specified on page 2, lines 3-4 are defined on page 2, lines 5-7. The fields contained in each header record belonging to the group defined on page 2, lines 8-9, are defined on page 2, lines 10-13. The fields contained in each header record belonging to the group defined on page 2, lines 14-15, are defined on page 2, line 16 and on page 3, lines 1-3.

Output Field Name (Columns 34-39)

None of the fields defined on page 2, lines 5-7, 10-13, and 16, and on page 3, lines 1-3 may be labeled, because a literal is to be placed in each of these fields at the time the object program is compiled. Under these circumstances, no entry is made in columns 34-39 on any of the affected lines.

Zero Suppression (Column 40)

A zero suppression operation is not to be performed on the contents of any of the fields defined on page 2, lines 5-7, 10-13, and 16, and on page 3, lines 1-3. As a result, no entry is made in column 44 on these lines.

Blank After (Column 41)

None of the fields defined on page 2, lines 5-7, 10-13, and 16, and on page 3, lines 1-3 are to be set to blanks after each record containing them is printed. As a result, no entry is made in column 41 on these lines.

Output Field End Position (Columns 42-45)

The last position in the fields defined on page 2, lines 5-7 coincides, respectively, with 57th, 78th and 88th position in each record that contains the field. As a result, 0057, 0078 and 0088 are entered in columns 42-45 on page 2, lines 5-7, respectively.

The last position in the fields defined on page 2, lines 10-13 coincides with the 25th, 80th, 97th, and 109th position in each record that contains the field. Therefore, 0025, 0080, 0097 and 0109 are entered in columns 42-45 on page 2, lines 10-13, respectively.

The last position in the fields defined on page 2, line 16, and on page 3, lines 1-3 coincides, respectively, with the 40th, 60th, 79th and 108th position in each record that contains the field. As a result, 0040, 0060, 0079 and 0108 are entered in columns 42-45 on page 2, line 16, and on page 3, lines 1-3, respectively.

Constant or Edit Word (Columns 46-70)

The contents of an RPG literal are to be placed in each of the affected fields at the time the object program is compiled. In this situation, the relevant literal is entered in columns 46-70 on page 2, lines 5-7, 10-13, and 16, and on page 3, lines 1-3 (see Figure 25).

APPENDIX A

Table Lookup Operations

Table lookup operations may be performed on a simple type of table. This table may not have more than one table function associated with each table argument. All arguments must be the same number of characters in length, and all functions must be the same number of characters in length. The arguments and functions that make up the table may be on the same input file or on different input files. When they appear on the same input file they must be placed on the file in logical pairs. That is, each table argument must immediately precede the table function with which it is associated, or each table argument must immediately follow the table function with which it is associated. If the arguments and functions that make up the table are contained in more than one physical record on an input file, the format of all these records must be the same. Table arguments may consist of alphameric characters or nothing but numeric characters. They may be in either random or sequential order. If they are in a sequential order, that order may be ascending or descending.

INDEX

ADD (Add), 24
Appendix A, 66
  Table Lookup Operations, 66
Ascending Sequence (Definition), 29
Blank After (Col. 41), 34,44,48,53 60,64
Blocking Factor (Col. 22-25), 16,40,50
Calculation Specifications Entries, 23, 53,58
  Conditions (Cols. 9-17), 23
  Control Level (Cols. 7-8), 23
  Factor 1 (Cols. 18-27), 24
  Factor 2 (Cols. 33-42), 26
  Operation Code (Cols. 28-32), 24
  Result Field Conditions (Cols. 54-59), 27
  Result Field Decimal Position (Col. 52), 27
  Result Field Half Adjust (Col. 53), 27
  Result Field Length (Cols. 49-51), 27
  Result Field Name (Cols. 43-48), 26
Coding Card, 11
Coding Form, 10
Columns 31-74 (File Defs.), 16
Columns 71-74 (Input Specs.), 23
Columns 71-74 (Output-Format Specs.), 36
Comments (Cols. 60-74: Calculation Specs.), 29
Comments Line, 14
Common Columns, 11
  Form Type (Col. 6), 14
  Line Number (Cols. 3-5), 13
  Page Number (Cols. 1-2), 11
  Program Ident. (Cols. 75-80), 14
COMP (Compare), 25
Conditions (Cols. 9-17), 23,50,58
Constant-Edit Word (Cols. 46-70), 35, 44,48,53,54,61,65
Control Break (Definition), 9
Control Field (Definition), 9
Control Level (Cols. 7-8), 23,50,58
Control Levels (Definition), 21
Descending Sequence (Definition), 29
Detail Record (Definition), 32
DIV (Divide), 25
DIVR (Divide With Remainder), 25
Edit Word (Definition), 35
Entries
  Calculation Specifications, 23
  File Definitions, 14
  Input Specifications, 16
  Output-Format Specifications, 29
Examples, 37
  Program One (80/80 Listing), 38
  Program Two (Detail Records), 44
  Program Three (Final Total), 48
  Program Four (Editing), 53
  Program Five (Minor Totals), 56
  Program Six (Header Records), 61
Factor 1 (Cols. 18-27), 24,51,58
Factor 2 (Cols. 33-42), 26,51,58
Field Condition (Definition), 22
Field Condition (Cols. 65-70), 22, 42,47
File Definitions Entries, 14,39
  Blocking Factor (Cols. 22-25), 16
  Field Sequencing (Cols. 19-20), 15
  File Designation (Col. 18), 15
  File Format (Col. 21), 15
  File Name (Cols. 7-16), 14
  File Type (Cols. 17), 15
  Record Length (Cols. 26-30), 16
File Designation (Col. 18), 15,40
File Format (Col. 21), 15,40,50
Field-Record Relation (Cols. 63-64), 22, 42,47
Field Sequencing (Cols. 19-20), 15,40
File Name (File Defs.), 14,39
File Name (Input Specs.), 17, 40
File Name (Output-Format Specs.), 30, 42,52,59,61
File Type (Col. 17), 15,40
First Page Special Indicator, 34
Form Type (Col. 6), 14
Header Record (Definition), 32
Input Field Control Level (Cols. 59-60), 21,42,46,58
Input Field Decimal Position (Col. 52), 21,42,46
Input Field Definition Entries, 21, 41,46
Input Field - From (Cols. 44-47), 21, 41,46
Input Field - To (Cols. 48-51), 21, 41,46
Input Field Name (Cols. 53-58), 21, 42,46
Input File (Sample Program), 37
Input Record Definition Entries, 17,40
Input Record Groups (Definition), 16
Input Specifications Entries, 16,40,58
  Field Indicator (Cols. 65-70), 22
  Field-Record Relation (Cols. 63-64), 22
  File Name (Cols. 7-16), 17
  Matching Field Code (Cols. 61-62), 22
  Number of Records (Col. 19), 19
  Option (Col. 20), 19
  Record Identification Codes (Cols. 23-43), 20
  Result Indicator (Cols. 21-22), 19
  Sequence (Cols. 17-18), 17
  Input Field Control Level (Cols. 59-60), 21
  Input Field Decimal Position (Col. 52), 21
  Input Field - From (Cols. 44-47), 21
  Input Field - To (Cols. 48-51), 21
  Input Field Name (Cols. 53-58), 21
Introduction, 5

Last Record Special Indicator, 34
Line Number (Cols. 3-5), 13
Line-Print Line, 8
LOKUP (Table Lookup), 26
Master File, 9
Matching Field Code (Cols. 61-62), 22, 42,47
Matching Field (Definition), 22
MOVE (Move), 26
MULT (Multiply), 25
MZONE (Move Zone), 26
Number of Records (Col. 19), 19,41
Object Program Logic, 8
Operation Code (Cols. 28-32), 24,51,58
Operation Codes, 24
  ADD (Add), 24
  COMP (Compare), 25
  DIV (Divide), 25
  DIVR (Divide With Remainder), 25
  LOKUP (Table Lookup), 26
  MOVE (Move), 26
  MULT (Multiply), 25
  MZONE (Move Zone), 26
  SUB (Subtract), 25
  Z-ADD (Zero and Add), 24
  Z-SUB (Zero and Subtract), 25
Option (Col. 20), 19,41
Output Conditions (Cols. 25-33), 33,43 52,60,64
Output Field Definition Entries, 34,43, 47,52,60,64
Output Field End Position (Cols. 42-45), 35,44,48,53,61,65
Output Field Name (Cols. 34-39), 34,43 47,52,60,64
Output-Format Specs. Entries, 29,51,59
  Blank After (Col. 41), 34
  Constant-Edit Word (Cols. 46-70), 35
  File Name (Columns 7-16), 30
  Output Conditions (Cols. 25-33), 33
  Output Field End Position (Cols. 42-45), 35
  Output Field Name (Cols. 34-39), 34
  Output Record Type (Col. 17), 30
  Printer Channel Skipping (Cols. 21-24), 33
  Printer Spacing (Columns 19-20), 33
  Stacker (Col. 18), 32
  Zero Suppression (Col. 40), 34
Output Record Definitions Entries, 30, 42,52,59,61
Output Record Groups (Definition), 29
Output Record Type (Col. 17), 30,43,52 59,61
Page Number (Cols. 1-2), 11
Page Numbering, 30
Page Overflow Special Indicator, 34
Printer Channel Skipping (Cols. 21-24), 33,43,52,60,64
Printer Spacing (Cols. 19-20), 33,43, 52,60,62
Program Five (Minor Totals), 56
  Input Specs. Entries, 58
  Calculation Specs. Entries, 58
  Output-Format Specs. Entries, 59
Program Four (Editing), 53
Program Identification (Cols. 75-80), 14
Program One (80/80 Listing), 38
  Input Specs. Entries, 40
  Output-Format Specs. Entries, 42
Program Six (Header Records), 61
Program Three (Final Total), 48
  File Defs. Entries, 50
  Calculation Specs. Entries, 50
  Output-Format Specs. Entries, 51
Program Two (Detail Records), 44
  File Defs. Entries, 45
  Input Specs. Entries, 46
  Output-Format Specs. Entries, 47
Random Order (Definition), 27
Record Available Result Indicators, 19
Record Field (Definition), 8
Record Identification Code (Definition), 20
Record Identification Codes (Cols. 23-43), 20,41
Record Length (Cols. 26-30), 16,40,50
Result Field (Definition), 8
Result Field Conditions (Cols. 54-59), 27,51,59
Result Field Decimal Position (Col. 52), 27,51,59
Result Field Half Adjust (Col. 53), 27, 51,59
Result Field Length (Cols. 49-51), 27, 51,59
Result Field Name (Cols. 43-48), 26, 51,58
Result Indicator (Cols. 21-22), 19,41
Result Indicators (Definition), 19
RPG (Report Program Generator)
  Coding Card, 11
  Coding Form, 10
  Elements Of, 6
  Language, 6
  Language Entries, 10
  Literal, 5
  Object Programs, Capabilities Of, 7
  Purpose Of, 5
Sequence (Cols. 17-18), 17,41
Spacing Chart, 30
Special Indicators (Definition), 33
Stacker (Col. 18), 32,43,52,59,62
SUB (Subtract), 25
Support Files, 9
Total Record (Definition), 32
Update Files, 15
Using the RPG, 10
Z-ADD (Zero and Add), 24
Zero Suppression (Col. 40), 34,43,47 53,60,64
Z-SUB (Zero and Subtract), 25

READERS' COMMENTS

IBM System/360 Operating System Report Program Generator, Form C28-6517-0

Your comments regarding this publication will help us improve future editions. Please comment on the usefulness and readability of the publication, suggest additions and deletions, and list specific errors and omissions.

USEFULNESS AND READABILITY

fold fold

SUGGESTED ADDITIONS AND DELETIONS

CUT ALONG LINE

ERRORS AND OMISSIONS (give page numbers)

fold fold

Name _____

Title or Position _____

Address _____

FOLD ON TWO LINES, STAPLE AND MAIL
No Postage Necessary if Mailed in U. S. A.

STAPLE

FOLD FOLD

FIRST CLASS
PERMIT NO. 81
POUGHKEEPSIE, N. Y.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN U. S. A.

POSTAGE WILL BE PAID BY
IBM CORPORATION
P. O. BOX 390
POUGHKEEPSIE, N. Y. 12602

ATTN: PROGRAMMING SYSTEMS PUBLICATIONS,
DEPT. D58

FOLD FOLD

CUT ALONG LINE

3/64:10M-VO-72

STAPLE STAPLE

Printed in U.S.A. C28-6517-0

IBM®

International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, N.Y. 10601